RUNTIME-ADAPTIVE TEMPORAL INTEGRATION IMPROVES SPARSITY AND ACCURACY IN SPIKING NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking neural networks inherently operate in an event-driven manner, yet their neurons typically rely on a fixed temporal integration scale during inference, regardless of changes in input conditions or their own activity states. Although some neuron models learn task-dependent membrane time constants during training, these time constants are typically kept fixed at inference, limiting the ability of SNNs to adapt their temporal integration to runtime conditions. Motivated by this observation, we study runtime-adaptive temporal integration as a computational perspective in which each neuron can adjust its temporal integration online according to local input conditions and recent firing activity. To instantiate this perspective, we develop the Runtime-Adaptive Temporal Integration Leaky Integrate-and-Fire (RATI-LIF) neuron, a lightweight LIF-based model that provides a practical realization of runtime-adaptive temporal integration. RATI-LIF learns a task-dependent reference integration timescale through layer-wise base time constants and further modulates this timescale at runtime through neuron-level input-driven and output-spike-driven dynamics. Experimental results show that RATI-LIF substantially reduces neuronal firing rates and estimated energy consumption while improving classification accuracy. Taken together, these results suggest that runtime-adaptive temporal integration offers a promising direction for building sparser and more effective spiking neural networks, with RATI-LIF providing a lightweight and practical realization of this idea.

1 INTRODUCTION

Spiking neural networks (SNNs) have attracted growing attention due to their event-driven operation and temporal coding capabilities (Li et al., 2021a; Zhou et al., 2021), which naturally support energy-efficient computation and the processing of dynamic, time-varying signals (Maass, 1997; Kasabov et al., 2013; Yu et al., 2020; Sun et al., 2022). A common observation in biological spiking systems is that neural computation typically operates under sparse and low-rate spiking activity, a regime closely associated with metabolic efficiency and stable information processing Attwell & Laughlin (2001); Lennie (2003); Laughlin (2001). These properties make SNNs particularly appealing for applications requiring rapid response and low power consumption, especially when integrated with neuromorphic hardware (Farsa et al., 2019; Wu et al., 2022).

Neuronal dynamics, describing the time-varying electrical behavior and state evolution of spiking neurons, form the computational foundation of spiking neural networks and shape their information processing capabilities. These dynamics are profoundly influenced by intrinsic cellular properties, among which the membrane time constant plays a pivotal role (Abbott & Dayan, 2001; Gerstner & Kistler, 2002; Izhikevich, 2007). The choice of membrane time constant therefore critically influences the temporal dynamics, responsiveness, and information processing capabilities of SNNs.

Despite its importance, the membrane time constant is treated as a fixed hyperparameter in many widely used SNN formulations, or learned during training but kept unchanged once inference begins (Wu et al., 2018; Fang et al., 2021b; Perez-Nieves et al., 2021). Consequently, the temporal integration scale of a neuron is predetermined before inference and cannot adapt to changes in the current input condition or the neuron's evolving activity state. This contrasts with biological and biophysi-

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.

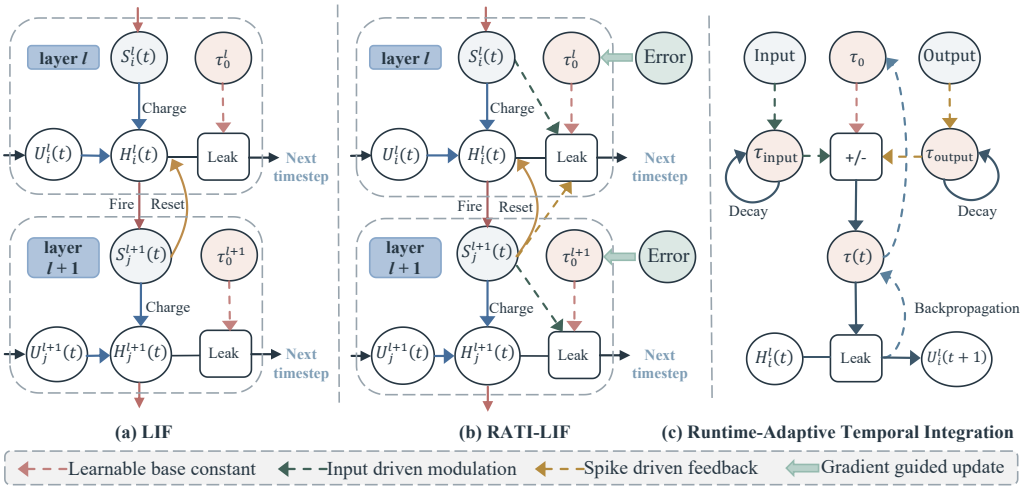

Figure 1: Overview of standard LIF and the proposed RATI-LIF. (a) Standard LIF uses a fixed membrane time constant during inference. (b) RATI-LIF uses a learnable base time constant as a reference timescale and adapts it at runtime using local neuronal signals. (c) In RATI-LIF, the layer-wise base time constant τ_0 is learned during training, while input-driven τ_{input} and spike-driven τ_{output} modulation jointly determine the neuron-level time constant $\tau(t)$, which controls membrane leakage at each timestep.

cal observations, where neuronal integration properties can vary with ongoing synaptic and intrinsic activity (Koch et al., 1996). In particular, increased synaptic conductance can shorten the effective membrane time constant and alter how rapidly membrane potentials respond to incoming signals (Kuhn et al., 2004), while activity-dependent intrinsic conductances can further regulate neuronal excitability over time (Ha & Cheong, 2017). These observations suggest that temporal integration need not be static during neural processing, motivating the question of whether spiking neurons should likewise adapt their integration timescale to runtime conditions.

Runtime-dependent temporal integration has emerged in different forms across existing neuronal models. In classical biophysical models, such as Hodgkin–Huxley formulations and conductance-based neurons, state-dependent ionic and synaptic conductances naturally give rise to varying effective temporal integration timescales (Hodgkin & Huxley, 1952; Destexhe, 1997; Gerstner & Kistler, 2002; Gütig & Sompolinsky, 2009). More recently, Liquid Time-Constant (LTC) networks introduce varying time constants coupled to the evolving network state, enabling adaptive temporal integration (Hasani et al., 2021; Yin et al., 2023). In spiking neural networks, the brain-inspired adaptive Leaky Integrate-and-Fire (BA-LIF) neuron further demonstrates that the membrane time constant can be adjusted according to runtime inputs (Zhang et al., 2025). Relatedly, recent work on dynamic conductance introduces activity-dependent membrane conductance, allowing neuronal dynamics to adapt to ongoing activity (Bai et al., 2026). Despite their different formulations and original motivations, these models point to a common computational behavior: the temporal integration of a neuron need not remain fixed during inference, but can adapt according to its runtime condition. We refer to this shared behavior as **runtime-adaptive temporal integration**. Existing studies have primarily investigated such adaptation within their respective model-specific mechanisms and objectives. To our knowledge, however, the relationship between this shared runtime-adaptive behavior and neuronal firing sparsity has received less direct attention. Motivated by this recurring pattern, we investigate whether adapting temporal integration according to locally available input and neuronal activity can promote sparse firing while maintaining effective temporal information processing.

To investigate this question, we develop RATI-LIF, a lightweight and explicit instantiation of Runtime-Adaptive Temporal Integration based on the Leaky Integrate-and-Fire neuron. Rather than operating with a fixed temporal integration scale throughout inference, RATI-LIF combines a task-dependent base integration timescale learned during training with runtime modulation driven by locally available neuronal signals. Specifically, a layer-wise learnable base time constant estab-

lishes a task-dependent reference timescale, which is further adjusted at the neuron level according to current inputs and recent spiking activity. In this way, RATI-LIF enables each neuron to adapt its temporal integration to its local runtime condition, providing a simple realization for examining whether runtime-adaptive temporal integration can promote sparse neuronal firing while preserving effective temporal information processing.

To evaluate runtime-adaptive temporal integration across temporal modalities, we conduct experiments on the EEG dataset DEAP (Koelstra et al., 2011), the neuromorphic vision benchmarks DVS-Gesture (Amir et al., 2017) and CIFAR10-DVS (Li et al., 2017), and the event-based speech benchmarks SHD and SSC. Across datasets and SNN architectures, RATI-LIF generally promotes sparser firing while maintaining competitive task performance, with reduced firing activity and improved accuracy in most settings. The reduced firing activity further demonstrates the potential for energy-efficiency benefits. Overall, these results provide empirical evidence that runtime-adaptive temporal integration can promote sparse firing while preserving effective spiking computation. The main contributions are threefold:

- We investigate **runtime-adaptive temporal integration** as a computational perspective for spiking neurons, in which the temporal integration scale adapts during inference according to local inputs and neuronal activity. In particular, we examine its relationship with neuronal firing sparsity while preserving effective temporal information processing.
- To investigate this perspective, we develop **RATI-LIF** as a lightweight instantiation combining a layer-wise learnable base time constant with neuron-level input- and output-spike-driven modulation, enabling temporal integration to adapt to local runtime conditions.
- Extensive experiments across physiological, neuromorphic vision, and event-based speech benchmarks demonstrate the applicability of RATI-LIF across temporal modalities and SNN architectures. In settings where firing activity is evaluated, RATI-LIF generally promotes sparser neuronal firing while maintaining competitive task performance, with improved classification accuracy observed in most cases. The resulting sparsification further suggests potential energy-efficiency benefits.

2 METHOD

We first revisit the standard Leaky Integrate-and-Fire (LIF) neuron as a static reference. We then introduce RATI-LIF, a lightweight instantiation of runtime-adaptive temporal integration, where a task-dependent reference timescale learned during training is modulated at runtime by local neuronal signals. Finally, we analyze how such adaptation affects neuronal firing and under what conditions it can promote a sparser firing regime. Additional biophysical motivation based on conductance-dependent neuronal dynamics is provided in Appendix A.1.

2.1 LIF NEURON WITH STATIC TEMPORAL INTEGRATION

The LIF neuron is widely used in spiking neural networks due to its simple and computationally efficient dynamics. Its temporal integration behavior is governed by a membrane decay factor that remains fixed throughout inference. As illustrated in Fig. 1(a), the neuron integrates incoming synaptic current $I^l(t)$ into its membrane potential while gradually dissipating previously accumulated information. The discrete-time neuronal dynamics are given by

$$I^l(t) = f\left(W^l, S^{l-1}(t)\right), \tag{1}$$

$$U^l(t) = \rho_{\mathrm{m}}\left(H^l(t-1) - S^l(t-1)V_{\mathrm{th}}\right), \tag{2}$$

$$H^l(t) = U^l(t) + I^l(t), \tag{3}$$

$$S^l(t) = \Theta\left(H^l(t) - V_{\mathrm{th}}\right) = \begin{cases} 1, & H^l(t) \geq V_{\mathrm{th}}, \\ 0, & H^l(t) < V_{\mathrm{th}}. \end{cases} \tag{4}$$

Here, $S^l(t)$ and $H^l(t)$ denote the output spike and membrane potential at time step t for the l-th layer, respectively. V_{th} is the firing threshold, $f(\cdot)$ denotes the convolutional or fully connected operation, and W^l represents the synaptic weights between adjacent layers.

The membrane decay factor ρ_{m} determines how much previously accumulated membrane potential is retained across time and is related to the membrane time constant τ by

$$\rho_{\mathrm{m}} = \exp\left(-\frac{\Delta t}{\tau}\right), \tag{5}$$

where Δt denotes the simulation timestep. Since τ, and consequently ρ_{m}, remains constant during inference, the contribution of past inputs is governed by a fixed temporal decay kernel. The standard LIF neuron therefore provides a static temporal integration reference against which we study runtime-adaptive temporal integration.

2.2 RATI-LIF: Runtime-Adaptive Temporal Integration

Rather than keeping the temporal integration scale fixed throughout inference, we allow it to vary according to the local runtime condition of each neuron. We describe this behavior using a time-varying membrane time constant

$$\tau_n^l(t) = F\left(\tau_0^l, I_n^l(t), S_n^l(t-1)\right), \tag{6}$$

where τ_0^l denotes a layer-wise learnable reference timescale, while $I_n^l(t)$ and $S_n^l(t-1)$ provide locally available runtime information for neuron n in layer l. In contrast to a membrane time constant that is learned during training but remains fixed once inference begins, as in PLIF (Fang et al., 2021b), Eq. (6) allows the temporal integration scale to evolve with the runtime neuronal state.

To provide a lightweight and explicit instantiation of this behavior, RATI-LIF introduces two local modulation states driven by the current input and recent output spikes:

$$\tau_{\mathrm{input},n}^l(t) = \lambda_i \tau_{\mathrm{input},n}^l(t-1) + \eta_i I_n^l(t), \tag{7}$$

$$\tau_{\mathrm{output},n}^l(t) = \lambda_o \tau_{\mathrm{output},n}^l(t-1) + \eta_o S_n^l(t-1), \tag{8}$$

where λ_i and λ_o control the temporal persistence of the two modulation states, while η_i and η_o determine their sensitivity to the corresponding neuronal signals. The runtime membrane time constant is then defined as

$$\tau_n^l(t) = \mathrm{clip}\left(\tau_0^l - \tau_{\mathrm{input},n}^l(t) - \tau_{\mathrm{output},n}^l(t), \tau_{\min}, \tau_{\max}\right). \tag{9}$$

Accordingly, the membrane decay factor becomes

$$\rho_{\mathrm{m},n}^l(t) = \exp\left(-\frac{\Delta t}{\tau_n^l(t)}\right), \tag{10}$$

and the membrane potential is updated as

$$U_n^l(t) = \rho_{\mathrm{m},n}^l(t)\left(H_n^l(t-1) - S_n^l(t-1)V_{\mathrm{th}}\right). \tag{11}$$

In this formulation, τ_0^l is a layer-wise learnable time constant optimized during training, which establishes a task-dependent reference timescale. The two local modulation states allow neurons to adjust this reference according to runtime conditions. RATI-LIF realizes runtime-adaptive temporal integration through local recursions without extra learnable parameters or control networks.

2.3 Theoretical Analysis of Firing Sparsity

We analyze RATI-LIF through a matched static reference that receives the same realized input-current sequence and uses the same firing threshold and subtractive reset rule, while fixing its membrane timescale to the learned reference τ_0. This comparison follows the classical interpretation of LIF neurons as leaky temporal integrators whose retained membrane state is governed by the membrane time constant (Gerstner et al., 2014). In contrast to learnable but inference-static temporal integration such as PLIF (Fang et al., 2021b), RATI-LIF further modulates the learned reference timescale according to runtime neuronal activity. The matched comparison therefore isolates the neuronal effect of runtime temporal adaptation under a fixed input process. For clarity, neuron and layer indices are omitted, $j \in \{\mathrm{A}, 0\}$ denotes the adaptive and static neurons, and all expectations are taken over the underlying input/data randomness and any neuronal randomness.

Define the post-spike residual membrane state as $R_j(t) = H_j(t) - V_{\mathrm{th}} S_j(t)$. The neuronal dynamics can then be written as

$$R_j(t) = \rho_{\mathrm{m},j}(t) R_j(t-1) + I(t) - V_{\mathrm{th}} S_j(t), \tag{12}$$

where $\rho_{\mathrm{m},0} = \exp(-\Delta t/\tau_0)$ and $\rho_{\mathrm{m},A}(t) = \exp(-\Delta t/\tau_{\mathrm{A}}(t))$.

For an observation horizon of T timesteps, let $\bar{r}_T^j = T^{-1} \sum_{t=1}^{T} \mathbb{E}[S_j(t)]$ denote the expected number of spikes per timestep and $\mu_{I,T} = T^{-1} \sum_{t=1}^{T} \mathbb{E}[I(t)]$ the mean input drive. We further define the signed temporal-decay contribution as

$$\mathcal{D}_T^j = \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\left(1 - \rho_{\mathrm{m},j}(t)\right) R_j(t-1) \right], \tag{13}$$

and the finite-horizon boundary term as $B_T^j = [\mathbb{E}[R_j(0)] - \mathbb{E}[R_j(T)]]/T$. The corresponding firing rate in physical units is $\bar{r}_T^j / \Delta t$. Summing Eq. (12) over time gives the exact identity

$$\boxed{V_{\mathrm{th}} \bar{r}_T^j = \mu_{I,T} - \mathcal{D}_T^j + B_T^j.} \tag{14}$$

This identity follows directly from the discrete neuronal dynamics and requires no specific distributional assumption on the input. More generally, firing statistics of integrate-and-fire neurons are governed by the interaction between synaptic drive, membrane dynamics, and threshold/reset mechanisms (Gerstner et al., 2014; Burkitt, 2006). Because the matched neurons receive the same current sequence, the input term cancels:

$$V_{\mathrm{th}} \left(\bar{r}_T^{\mathrm{A}} - \bar{r}_T^0 \right) = - \left(\mathcal{D}_T^{\mathrm{A}} - \mathcal{D}_T^0 \right) + \Delta B_T, \qquad \Delta B_T = B_T^{\mathrm{A}} - B_T^0. \tag{15}$$

Hence,

$$\boxed{\bar{r}_T^{\mathrm{A}} \le \bar{r}_T^0 \quad \Longleftrightarrow \quad \mathcal{D}_T^{\mathrm{A}} - \mathcal{D}_T^0 \ge \Delta B_T.} \tag{16}$$

To relate this condition to temporal adaptation, let $\alpha(\tau) = 1 - \exp(-\Delta t/\tau)$ and $a_0 = \alpha(\tau_0)$, and define the mean adaptive decay coefficient as $a_T = T^{-1} \sum_{t=1}^{T} \mathbb{E}[\alpha(\tau_{\mathrm{A}}(t))]$. Since $\alpha(\tau)$ decreases with τ and is convex for $\tau \ge \Delta t/2$, the average temporal contraction

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[\tau_{\mathrm{A}}(t)] \le \tau_0 \tag{17}$$

implies $a_T \ge a_0$ when $\tau_{\mathrm{A}}(t) \ge \Delta t/2$ almost surely. The same ordering follows directly if $\tau_{\mathrm{A}}(t) \le \tau_0$ pointwise. The monotonicity and Jensen arguments are given in Appendix A.2.

An increased mean decay coefficient alone does not guarantee reduced firing, because runtime adaptation can also alter the residual membrane statistics. As derived in Appendix A.2, let $m_{0,T} = T^{-1} \sum_{t=1}^{T} \mathbb{E}[R_0(t-1)]$ denote the mean residual state of the static reference. Let ε_T, κ_T, and b_T bound, respectively, the adaptive–static residual-state mismatch, an adverse decay–state covariance, and the finite-horizon boundary contribution. If $m_{0,T} \ge 0$ and

$$\boxed{(a_T - a_0) m_{0,T} \ge a_T \varepsilon_T + \kappa_T + b_T \quad \Longrightarrow \quad \bar{r}_T^{\mathrm{A}} \le \bar{r}_T^0} \tag{18}$$

Eq. (18) shows that the increase in temporal decay must be large enough to dominate changes in the residual operating point, adverse decay–state correlation, and finite-horizon effects. The exact moment decomposition, boundary control, stationary specialization, and a stronger pathwise result under nonnegative inputs are provided in Appendix A.2.

This result characterizes matched neurons under the same input current. In multilayer SNNs, neuronal inputs are further shaped by the dynamics of upstream layers.

3 Experiments

We evaluate RATI-LIF across neuromorphic vision, physiological signals, and event-based speech benchmarks, focusing on firing sparsity and task performance, energy efficiency, cross-modal generalization, and component contributions. Unless otherwise specified, results are averaged over three independent runs. Detailed dataset descriptions, preprocessing, training configurations, and hyperparameter settings are provided in Appendix A.4.

Table 1: Comparison of LIF, PLIF, and our RATI-LIF neuron on neuromorphic datasets. PLIF and RATI-LIF introduce only one learnable time-constant parameter per neuronal layer; therefore, their additional parameter overhead is negligible and does not alter the reported parameter count when rounded to two decimal places in millions (M). RATI-LIF results are highlighted with gray shading.

Dataset	Method	Network	Neuron	T	Params ↓	FR (%) ↓	Acc. (%) ↑
DVS-Gesture	STBP-tdBN (Zheng et al., 2021)	ResNet-17	LIF	40	11.18 M	**4.02**	96.87
			PLIF		11.18 M	4.70	97.22
			RATI-LIF		11.18 M	4.35	**97.56**
	SEWResNet (Fang et al., 2021a)	SEW ResNet	PLIF	16	0.13 M	5.50	97.92
			RATI-LIF		0.13 M	**4.48**	**98.61**
	SLTT (Meng et al., 2023)	VGG-11	LIF	20	123.24 M	1.93	97.92
			PLIF		123.24 M	1.63	98.26
			RATI-LIF		123.24 M	**1.14**	**98.61**
	Spikformer (Zhou et al., 2022)	Spikingformer	LIF	16	2.57 M	4.26	98.30
			PLIF		2.57 M	4.31	98.61
			RATI-LIF		2.57 M	**3.83**	**98.96**
	QKFormer (Zhou et al., 2024)	Spikingformer	LIF	16	1.50 M	**6.14**	98.60
			PLIF		1.50 M	17.27	98.96
			RATI-LIF		1.50 M	13.98	**99.30**
CIFAR10-DVS	SEWResNet (Fang et al., 2021a)	SEW ResNet	PLIF	16	1.19 M	19.37	74.40
			RATI-LIF		1.19 M	**13.68**	**76.40**
	SLTT (Meng et al., 2023)	VGG-11	LIF	10	123.24 M	8.05	76.40
			PLIF		123.24 M	7.50	**77.20**
			RATI-LIF		123.24 M	**5.63**	76.90
	Spikformer (Zhou et al., 2022)	Spikingformer	LIF	16	2.57 M	10.00	80.90
			PLIF		2.57 M	11.45	81.27
			RATI-LIF		2.57 M	**8.94**	**81.49**
	QKFormer (Zhou et al., 2024)	Spikingformer	LIF	16	1.50 M	**14.52**	84.00
			PLIF		1.50 M	19.33	84.20
			RATI-LIF		1.50 M	18.21	**86.60**

3.1 Accuracy and Firing Sparsity under Runtime-Adaptive Integration

We first evaluate the central hypothesis of this work on DVS-Gesture (Amir et al., 2017) and CIFAR10-DVS (Li et al., 2017). To isolate the effect of runtime-adaptive temporal integration across different network designs, we replace only the neuronal model in several representative SNN frameworks while otherwise preserving their corresponding network configurations. In particular, PLIF (Fang et al., 2021b) provides a natural inference-time static reference: its membrane time constant is learned during training but remains fixed during inference, whereas RATI-LIF further modulates the learned reference timescale according to runtime neuronal activity.

The selected methods span representative SNN training schemes and architectural families. We evaluate SLTT with VGG-11, STBP-tdBN with ResNet-17, SEWResNet with a SEW-ResNet backbone, and the Transformer-based Spikformer and QKFormer with their corresponding Spikingformer architectures. Together, these methods provide convolutional, residual, and spiking-transformer settings for evaluating RATI-LIF.

As shown in Tab. 1, RATI-LIF consistently reduces neuronal firing relative to its matched PLIF counterpart across all evaluated configurations on both datasets. The relative reduction in firing rate ranges from approximately 5.8% to 30.1%, covering convolutional, residual, and spiking-

Table 2: Comparison of energy consumption per timestep on the DEAP dataset. Arrows indicate the relative percentage increase (↑) or decrease (↓) compared to the standard LIF neuron. Accuracy improvements are reported as absolute differences (percentage points). RATI-LIF results are highlighted with gray shading.

Model	Firing rate (%)	Power (mJ)	Accuracy (%)
LIF	46.68	0.441	79.06
LTC-LIF (Yin et al., 2023)	25.42 (↓45.5%)	0.852 (↑93.2%)	85.86 (+6.80)
BA-LIF (Zhang et al., 2025)	48.56 (↑4.0%)	2.801 (↑535.2%)	86.88 (+7.82)
RATI-LIF (Ours)	**12.79 (↓72.6%)**	**0.122 (↓72.3%)**	**90.11 (+11.05)**

transformer architectures. At the same time, RATI-LIF improves classification accuracy in eight of nine settings, with only a minor decrease of 0.30 percentage points for SLTT on CIFAR10-DVS.

These results are consistent with the analysis in Sec. 2.3: introducing runtime adaptation on top of a learned static reference can shift neuronal activity toward a sparser firing regime while largely preserving, and often improving, task performance. Although the network-level experiments do not enforce the identical-current assumption used in the matched-neuron analysis, the consistent reduction from PLIF to RATI-LIF across architectures provides empirical evidence that a similar sparsification pattern also emerges in trained multilayer SNNs.

3.2 Energy Efficiency

Having established that runtime adaptation consistently reduces firing relative to the matched inference-time static reference, we next examine whether this reduction in neuronal activity translates into lower estimated energy consumption. Following prior SNN energy analyses (Li et al., 2021b; Rathi & Roy, 2021; Zhang et al., 2025), we estimate operation-level energy on the DEAP dataset using a common two-layer fully connected network, thereby reducing the influence of architectural differences. The energy model follows 45 nm CMOS estimates, where a multiply operation consumes 3.7 pJ and an accumulation operation consumes 0.9 pJ (Horowitz, 2014).

As shown in Tab. 2, RATI-LIF exhibits a firing rate of 12.79%, substantially lower than LIF (46.68%), LTC-LIF (25.42%), and BA-LIF (48.56%) under the evaluated setting. Although runtime modulation introduces additional local operations, the reduction in spike activity results in an estimated energy consumption of 0.122 mJ, compared with 0.441 mJ for standard LIF. RATI-LIF therefore reduces estimated energy by 72.3% relative to LIF while simultaneously improving accuracy from 79.06% to 90.11%. This result suggests that the sparsification associated with runtime-adaptive temporal integration can further translate into energy-efficiency benefits.

3.3 Generalization across Temporal Modalities

Physiological temporal signals. We next evaluate whether runtime-adaptive temporal integration generalizes beyond neuromorphic vision to physiological signals. We conduct valence and arousal classification on the DEAP dataset (Koelstra et al., 2011), following the preprocessing protocol of Zheng et al. (2024). Detailed preprocessing and training settings are provided in Appendix A.4.

As shown in Tab. 3, RATI-LIF achieves 92.35% and 94.31% accuracy on binary valence and arousal classification, respectively, and 90.11% and 92.57% on the corresponding three-class tasks. Compared with BISNN (Sun et al., 2025), RATI-LIF achieves the highest arousal accuracy in both settings, improving by 0.58 and 1.03 percentage points in binary and three-class classification, respectively, while BISNN achieves the highest valence accuracy by margins of 0.89 and 0.61 percentage points. Together with the neuromorphic results, these findings indicate that the benefits associated with runtime-adaptive temporal integration, as instantiated by RATI-LIF, extend beyond event-based vision to substantially different forms of temporal information.

Event-based speech. Beyond physiological signals, we further evaluate RATI-LIF on the SHD and SSC event-based speech benchmarks (Cramer et al., 2020) to examine its applicability to long temporal sequences. We consider feedforward, recurrent, and explicit-delay settings, thereby testing

Table 3: Performance comparison on DEAP under binary and three-class valence/arousal classification settings. "–" denotes unavailable numerical results.

Method	2 - class		3 - class	
	Val. (%)	Aro. (%)	Val. (%)	Aro. (%)
Transfer learning (Yan et al., 2022)	82.75	84.22	–	–
EEGNet (Gong et al., 2023)	76.82	81.14	–	–
SGLNet (Gong et al., 2023)	89.41	90.01	–	–
CNN (Islam et al., 2021)	78.22	74.92	70.23	70.25
DH-SNN (Zheng et al., 2024)	–	–	77.46	80.23
BSNN (Sun et al., 2025)	93.21	93.38	81.03	82.60
BISNN (Sun et al., 2025)	**93.24**	93.73	**90.72**	91.54
RATI-LIF (Ours)	92.35	**94.31**	90.11	**92.57**

Table 4: Comparison of different spiking neural networks on speech datasets. The best results are highlighted in bold. Baseline results are taken from the corresponding original papers, except for SNNdelays*, which is reproduced using public code.

Dataset	Method	Rec.	Delays	Hid. Layers	Params	Acc. (%)
SHD	LIF + aug (Cramer et al., 2020)	×	×	128	0.09 M	49.70
	TC-LIF (Zhang et al., 2024)	×	×	128-128	0.11 M	83.08
	RATI-LIF (Ours)	×	×	128	0.09 M	81.14
	RATI-LIF (Ours)	×	×	128-128	0.11 M	**89.36**
	TC-LIF (Zhang et al., 2024)	✓	×	128-128	0.14 M	88.91
	ALIF (Yin et al., 2020)	✓	×	128-128	0.14 M	84.40
	RadLIF (Bittar & Garner, 2022)	✓	×	128-128	0.14 M	**92.88**
	RATI-LIF (Ours)	✓	×	128-128	0.14 M	90.50
	Adaptive Delays (Pengfei et al., 2023)	×	✓	128-128	0.11 M	92.45
	Event-based delay (Mészáros et al., 2025)	×	✓	256-256	0.49 M	81.70
	SNNdelays* (Hammouamri et al., 2024)	×	✓	256-256	0.20 M	94.16
	RATI-LIF (Ours)	×	✓	256-256	0.20 M	**95.50**
SSC	LIF (Cramer et al., 2020)	×	×	128-128	0.11 M	38.50
	TC-LIF (Zhang et al., 2024)	×	×	128-128	0.11 M	63.46
	RATI-LIF (Ours)	×	×	128-128	0.11 M	**66.77**
	TC-LIF (Zhang et al., 2024)	✓	×	128	0.11 M	61.09
	RLIF (Bittar & Garner, 2022)	✓	×	128-128	0.14 M	73.87
	RadLIF (Bittar & Garner, 2022)	✓	×	128-128	0.14 M	73.25
	RATI-LIF (Ours)	✓	×	128-128	0.14 M	**76.23**
	Event-based delay (Mészáros et al., 2025)	×	✓	1024-1024	3.57 M	76.20
	SNNdelays* (Hammouamri et al., 2024)	×	✓	512-512	1.40 M	80.16
	RATI-LIF (Ours)	×	✓	512-512	1.40 M	**80.35**

the proposed neuron under a different temporal modality and several forms of temporal network structure. We do not report firing rates, as they are not consistently reported in prior speech studies.

As shown in Tab. 4, RATI-LIF consistently delivers strong performance across feedforward, recurrent, and explicit-delay configurations. In the feedforward setting, it achieves 89.36% on SHD and 66.77% on SSC, outperforming the corresponding non-recurrent baselines. With explicit delays, RATI-LIF further improves to 95.50% and 80.35%, respectively, achieving the best results on both datasets. In recurrent configurations, it remains competitive under the same hidden dimensions and

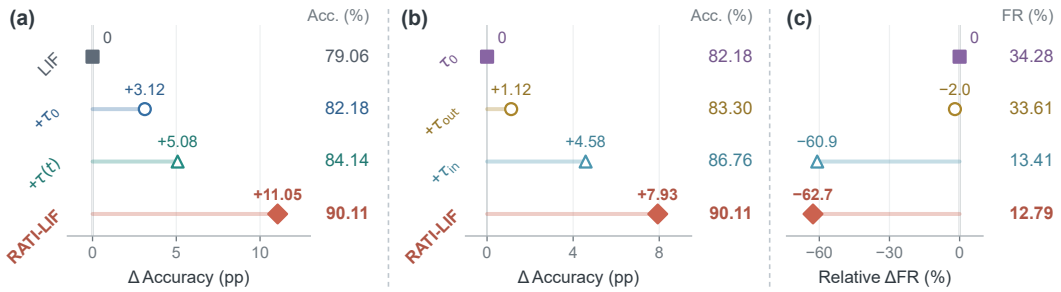

Figure 2: Ablation studies on DEAP. (a) Accuracy effects of the learnable reference timescale τ_0 and runtime adaptation $\tau(t)$. (b) Contributions of output-side (τ_{out}) and input-side (τ_{in}) runtime adaptation. (c) Firing rates (FR) under different configurations. Δ denotes the change relative to the baseline in each panel and "pp" denotes percentage points; left annotations show these changes, while right values report absolute accuracy or FR.

parameter count as RadLIF, while attaining 76.23% on SSC. These results highlight the effectiveness and architectural flexibility of RATI-LIF across different temporal processing paradigms.

3.4 Ablation and Mechanistic Analysis

Finally, we conduct ablation studies to examine how RATI-LIF components contribute to performance and sparsity gains. We first disentangle the contributions of the learnable reference timescale and runtime adaptation on DEAP. The same ablation on DVS-Gesture is provided in Appendix A.5 to test whether the trend generalizes to another temporal modality.

As shown in Fig. 2(a), both components improve over the fixed-timescale LIF baseline. Here, $+\tau_0$ and $+\tau(t)$ denote the learnable reference timescale and runtime adaptation, respectively, while RATI-LIF combines both. Learning a static reference timescale improves accuracy from 79.06% to 82.18%, while runtime adaptation with a fixed reference timescale reaches 84.14%.

We next examine the two local signals used for runtime modulation. Using τ_0 as the baseline, $+\tau_{\text{in}}$ and $+\tau_{\text{out}}$ selectively enable input-driven and output-spike-driven modulation, respectively.

As shown in Fig. 2(b), output-spike-driven modulation improves accuracy from 82.18% to 83.30%, whereas input-driven modulation increases it to 86.76%. Their combination further improves accuracy to 90.11%. Correspondingly, Fig. 2(c) shows that output-spike-driven modulation reduces the firing rate from 34.28% to 33.61%, while input-driven modulation produces a substantially stronger reduction to 13.41%. Combining both yields the lowest firing rate of 12.79%. Thus, input-driven modulation accounts for most of the sparsification, while output-spike-driven modulation provides an additional gain, with their combination improving both accuracy and sparsity.

Taken together, these ablations support complementary contributions from the task-dependent reference timescale and runtime modulation. The reference timescale provides a task-dependent integration scale, while input- and output-spike-driven signals jointly adapt it to runtime neuronal conditions.

4 Conclusion

We study runtime-adaptive temporal integration as a computational perspective for spiking neural networks and develop RATI-LIF as a lightweight realization combining a learnable reference timescale with runtime input- and output-spike-driven modulation. Our theoretical analysis characterizes conditions under which runtime adaptation can promote sparser firing relative to a matched static reference. Across neuromorphic vision, physiological signals, and event-based speech, RATI-LIF consistently reduces firing relative to matched PLIF counterparts while largely preserving or improving accuracy, supports energy-efficient computation through reduced spike activity, and demonstrates applicability across temporal modalities and network structures.

AI USE STATEMENT

Generative AI tools were used to assist with writing and literature discovery, and to support the development and verification of mathematical arguments in the "Theoretical Analysis of Firing Sparsity" section. All AI-assisted content, references, derivations, and proof steps were independently reviewed and verified by the authors, who take full responsibility for the final manuscript.

REFERENCES

Laurence F Abbott and Peter Dayan. Theoretical neuroscience. *Comput Math Model Neural*, 60: 489–95, 2001.

Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, et al. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7243–7252, 2017.

David Attwell and Simon B Laughlin. An energy budget for signaling in the grey matter of the brain. *Journal of Cerebral Blood Flow & Metabolism*, 21(10):1133–1145, 2001.

Qianyi Bai, Haiteng Wang, and Qiang Yu. A brain-inspired gating mechanism unlocks robust computation in spiking neural networks. volume 2026, pp. 156097–156129, April 2026.

Alexandre Bittar and Philip N Garner. A surrogate gradient spiking baseline for speech command recognition. *Frontiers in Neuroscience*, 16:865897, 2022.

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

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

Alain Destexhe. Conductance-based integrate-and-fire models. *Neural Computation*, 9(3):503–514, 1997.

Jianhao Ding, Zhiyu Pan, Yujia Liu, Zhaofei Yu, and Tiejun Huang. Robust stable spiking neural networks. *arXiv preprint arXiv:2405.20694*, 2024a.

Yongqi Ding, Lin Zuo, Mengmeng Jing, Pei He, and Yongjun Xiao. Shrinking your timestep: Towards low-latency neuromorphic object recognition with spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11811–11819, 2024b.

Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Advances in neural information processing systems*, 34:21056–21069, 2021a.

Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2661–2671, 2021b.

Edris Zaman Farsa, Arash Ahmadi, Mohammad Ali Maleki, Morteza Gholami, and Hima Nikafshan Rad. A low-cost high-speed neuromorphic hardware based on spiking neural network. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 66(9):1582–1586, 2019.

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

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.

Peiliang Gong, Pengpai Wang, Yueying Zhou, and Daoqiang Zhang. A spiking neural network with adaptive graph convolution and lstm for eeg-based brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 31:1440–1450, 2023.

Robert Gütig and Haim Sompolinsky. Time-warp–invariant neuronal processing. *PLoS biology*, 7 (7):e1000141, 2009.

Go Eun Ha and Eunji Cheong. Spike frequency adaptation in neurons of the central nervous system. *Experimental neurobiology*, 26(4):179, 2017.

Ilyass Hammouamri, Ismail Khalfaoui Hassani, and Timothée Masquelier. Learning delays in spiking neural networks using dilated convolutions with learnable spacings. In *International conference on learning representations*, volume 2024, pp. 17890–17903, 2024.

Ramin Hasani, Mathias Lechner, Alexander Amini, Daniela Rus, and Radu Grosu. Liquid time-constant networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7657–7666, 2021.

Hagar Hendy and Cory Merkel. Review of spike-based neuromorphic computing for brain-inspired vision: biology, algorithms, and hardware. *Journal of Electronic Imaging*, 31(1):010901–010901, 2022.

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, 1952.

Mark Horowitz. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE international solid-state circuits conference digest of technical papers (ISSCC)*, pp. 10–14. IEEE, 2014.

Md Rabiul Islam, Md Milon Islam, Md Mustafizur Rahman, Chayan Mondal, Suvojit Kumar Singha, Mohiuddin Ahmad, Abdul Awal, Md Saiful Islam, and Mohammad Ali Moni. Eeg channel correlation based model for emotion recognition. *Computers in Biology and Medicine*, 136:104757, 2021.

Eugene M Izhikevich. *Dynamical systems in neuroscience*. MIT press, 2007.

Nikola Kasabov, Kshitij Dhoble, Nuttapod Nuntalid, and Giacomo Indiveri. Dynamic evolving spiking neural networks for on-line spatio-and spectro-temporal pattern recognition. *Neural Networks*, 41:188–201, 2013.

Christof Koch, Moshe Rapp, and Idan Segev. A brief history of time (constants). *Cerebral cortex*, 6 (2):93–101, 1996.

Sander Koelstra, Christian Muhl, Mohammad Soleymani, Jong-Seok Lee, Ashkan Yazdani, Touradj Ebrahimi, Thierry Pun, Anton Nijholt, and Ioannis Patras. Deap: A database for emotion analysis; using physiological signals. *IEEE transactions on affective computing*, 3(1):18–31, 2011.

Alexandre Kuhn, Ad Aertsen, and Stefan Rotter. Neuronal integration of synaptic input in the fluctuation-driven regime. *Journal of Neuroscience*, 24(10):2345–2356, 2004.

Simon B Laughlin. Energy as a constraint on the coding and processing of sensory information. *Current opinion in neurobiology*, 11(4):475–480, 2001.

Peter Lennie. The cost of cortical computation. *Current biology*, 13(6):493–497, 2003.

Chen Li, Runze Chen, Christoforos Moutafis, and Steve Furber. Robustness to noisy synaptic weights in spiking neural networks. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2020.

Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. Cifar10-dvs: an event-stream dataset for object classification. *Frontiers in neuroscience*, 11:244131, 2017.

Sixu Li, Zhaomin Zhang, Ruixin Mao, Jianbiao Xiao, Liang Chang, and Jun Zhou. A fast and energy-efficient snn processor with adaptive clock/event-driven computation scheme and online learning. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 68(4):1543–1552, 2021a.

Yuhang Li, Yufei Guo, Shanghang Zhang, Shikuang Deng, Yongqing Hai, and Shi Gu. Differentiable spike: Rethinking gradient-descent for training spiking neural networks. *Advances in neural information processing systems*, 34:23426–23439, 2021b.

Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

Qingyan Meng, Mingqing Xiao, Shen Yan, Yisen Wang, Zhouchen Lin, and Zhi-Quan Luo. Towards memory-and time-efficient backpropagation for training spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6166–6176, 2023.

Balázs Mészáros, James C Knight, and Thomas Nowotny. Efficient event-based delay learning in spiking neural networks. *Nature Communications*, 16(1):10422, 2025.

S Pengfei, E Eqlimi, C Yansong, P Devos, and D Botteldooren. Adaptive axonal delays in feedforward spiking neural networks for accurate spoken word recognition, 2023.

Nicolas Perez-Nieves, Vincent CH Leung, Pier Luigi Dragotti, and Dan FM Goodman. Neural heterogeneity promotes robust learning. *Nature communications*, 12(1):5791, 2021.

Nitin Rathi and Kaushik Roy. Diet-snn: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 34(6):3174–3182, 2021.

Congyi Sun, Haohan Sun, Jin Xu, Jianing Han, Xinyuan Wang, Xinyu Wang, Qinyu Chen, Yuxiang Fu, and Li Li. An energy efficient stdp-based snn architecture with on-chip learning. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 69(12):5147–5158, 2022.

Hongze Sun, Shifeng Mao, Wuque Cai, Yan Cui, Duo Chen, Dezhong Yao, and Daqing Guo. Bisnn: bio-information-fused spiking neural networks for enhanced eeg-based emotion recognition. *Cognitive Neurodynamics*, 19(1):1–12, 2025.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018.

Yujie Wu, Rong Zhao, Jun Zhu, Feng Chen, Mingkun Xu, Guoqi Li, Sen Song, Lei Deng, Guanrui Wang, Hao Zheng, et al. Brain-inspired global-local learning incorporated with neuromorphic computing. *Nature Communications*, 13(1):65, 2022.

Zhanglu Yan, Jun Zhou, and Weng-Fai Wong. Eeg classification with spiking neural network: Smaller, better, more energy efficient. *Smart Health*, 24:100261, 2022.

Bojian Yin, Federico Corradi, and Sander M Bohté. Effective and efficient computation with multiple-timescale spiking recurrent neural networks. In *International Conference on Neuromorphic Systems 2020*, pp. 1–8, 2020.

Bojian Yin, Federico Corradi, and Sander M Bohté. Accurate online training of dynamical spiking neural networks through forward propagation through time. *Nature Machine Intelligence*, 5(5): 518–527, 2023.

Qiang Yu, Shenglan Li, Huajin Tang, Longbiao Wang, Jianwu Dang, and Kay Chen Tan. Toward efficient processing and learning with spikes: New approaches for multispike learning. *IEEE transactions on cybernetics*, 52(3):1364–1376, 2020.

Jiqing Zhang, Malu Zhang, Yuanchen Wang, Qianhui Liu, Baocai Yin, Haizhou Li, and Xin Yang. Spiking neural networks with adaptive membrane time constant for event-based tracking. *IEEE Transactions on Image Processing*, 2025.

Shimin Zhang, Qu Yang, Chenxiang Ma, Jibin Wu, Haizhou Li, and Kay Chen Tan. Tc-lif: A two-compartment spiking neuron model for long-term sequential modelling. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 16838–16847, 2024.

Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11062–11070, 2021.

Hanle Zheng, Zhong Zheng, Rui Hu, Bo Xiao, Yujie Wu, Fangwen Yu, Xue Liu, Guoqi Li, and Lei Deng. Temporal dendritic heterogeneity incorporated with spiking neural networks for learning multi-timescale dynamics. *Nature Communications*, 15(1):277, 2024.

Chenlin Zhou, Han Zhang, Zhaokun Zhou, Liutao Yu, Liwei Huang, Xiaopeng Fan, Li Yuan, Zhengyu Ma, Huihui Zhou, and Yonghong Tian. Qkformer: Hierarchical spiking transformer using qk attention. *Advances in Neural Information Processing Systems*, 37:13074–13098, 2024.

Shibo Zhou, Xiaohua Li, Ying Chen, Sanjeev T Chandrasekaran, and Arindam Sanyal. Temporal-coded deep spiking neural network with easy training and robust performance. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11143–11151, 2021.

Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. *arXiv preprint arXiv:2209.15425*, 2022.

A Appendix

A.1 Biophysical Motivation for Runtime-Adaptive Temporal Integration

This appendix provides additional biophysical motivation for runtime-adaptive temporal integration. The derivation is not intended to imply that RATI-LIF directly reproduces a particular conductance-based neuron model. Rather, conductance-based dynamics provide a concrete example in which the effective temporal integration timescale of a neuron varies according to its runtime state.

A.1.1 Effective Membrane Time Constant in Conductance-Based Neurons

The standard LIF neuron is commonly formulated as a current-based neuron in which synaptic inputs are injected as additive currents and the membrane time constant remains fixed. In contrast, conductance-based neuron models explicitly describe neuronal inputs and intrinsic dynamics through time-varying membrane conductances (Hodgkin & Huxley, 1952; Destexhe, 1997; Gütig & Sompolinsky, 2009; Bai et al., 2026).

A representative conductance-based membrane equation can be written as

$$\begin{aligned} C_m \frac{dU(t)}{dt} = &-g_L \left(U(t) - E_L\right) \\ &- g_{\mathrm{dyn}}(t) \left(U(t) - E_{\mathrm{dyn}}\right) + I_{\mathrm{ext}}(t), \end{aligned} \tag{19}$$

where C_m denotes the membrane capacitance, g_L is the passive leak conductance, and $g_{\mathrm{dyn}}(t)$ represents additional time-varying conductance components arising from synaptic or intrinsic neuronal dynamics. E_L and E_{dyn} denote the corresponding reversal potentials.

Considering the total instantaneous membrane conductance,

$$g_{\mathrm{tot}}(t) = g_L + g_{\mathrm{dyn}}(t), \tag{20}$$

the effective membrane time constant can be expressed as

$$\tau_{\mathrm{eff}}(t) = \frac{C_m}{g_L + g_{\mathrm{dyn}}(t)}. \tag{21}$$

Defining the passive membrane time constant as

$$\tau_0 = \frac{C_m}{g_L}, \tag{22}$$

Eq. (21) can equivalently be written as

$$\tau_{\mathrm{eff}}(t) = \frac{1}{\dfrac{1}{\tau_0} + \dfrac{g_{\mathrm{dyn}}(t)}{C_m}}. \tag{23}$$

Eq. (23) shows that the effective temporal integration scale of a conductance-based neuron need not remain constant. Instead, changes in the neuronal conductance state directly modify the rate at which previously accumulated membrane information decays. In particular, an increase in total membrane conductance decreases the effective membrane time constant and shortens the corresponding integration timescale (Kuhn et al., 2004).

A.1.2 Runtime Dependence of the Effective Integration Timescale

The dynamic conductance term in Eq. (19) can itself evolve according to ongoing neuronal and synaptic states. Without committing to a specific biophysical mechanism, we may express this dependence abstractly as

$$g_{\mathrm{dyn}}(t) = \mathcal{G}\left(\mathcal{X}(t)\right), \tag{24}$$

where $\mathcal{X}(t)$ denotes the locally available runtime state that influences membrane conductance, such as synaptic activity or intrinsic activity-dependent variables.

Substituting Eq. (24) into Eq. (23) yields

$$\tau_{\mathrm{eff}}(t) = \frac{1}{\dfrac{1}{\tau_0} + \dfrac{\mathcal{G}\left(\mathcal{X}(t)\right)}{C_m}}. \tag{25}$$

This formulation makes explicit that the effective integration timescale can depend on the runtime neuronal state. Recent work on dynamic membrane conductance similarly demonstrates that activity-dependent conductance can continuously regulate neuronal dynamics and information flow (Bai et al., 2026).

The above observation motivates the more general computational abstraction

$$\tau(t) = F\left(\tau_0, \mathcal{X}(t)\right), \tag{26}$$

where τ_0 defines a reference temporal scale and $\mathcal{X}(t)$ represents runtime information available to the neuron.

RATI-LIF adopts this computational abstraction without explicitly simulating the underlying conductance dynamics. Instead, it uses the current neuronal input and recent output-spiking activity as lightweight local signals for directly modulating the membrane time constant. This yields the runtime-adaptive formulation described in Sec. 2.2, while avoiding the need to explicitly maintain and solve detailed conductance-based neuronal dynamics.

A.2 Additional Analysis of Expected Firing Rate

A.2.1 Exact Finite-Horizon Firing-Rate Balance

Under the assumptions stated in Section 2.3, the residual dynamics in Eq. (12) can be rearranged as

$$\begin{aligned} V_{\mathrm{th}} S_j(t) = I(t) - \left(1 - \rho_{\mathrm{m},j}(t)\right) R_j(t-1) \\ + R_j(t-1) - R_j(t). \end{aligned} \tag{27}$$

Summing over $t = 1, \ldots, T$ gives

$$\begin{aligned} V_{\mathrm{th}} \sum_{t=1}^{T} S_j(t) = \sum_{t=1}^{T} I(t) - \sum_{t=1}^{T} \left(1 - \rho_{\mathrm{m},j}(t)\right) R_j(t-1) \\ + R_j(0) - R_j(T), \end{aligned} \tag{28}$$

where the residual-state differences telescope. Taking expectation and dividing by T yields

$$V_{\mathrm{th}} \bar{r}_T^j = \mu_{I,T} - \mathcal{D}_T^j + B_T^j, \tag{29}$$

which is Eq. (14) in the main text.

The required expectations exist at finite times because $0 < \rho_{\mathrm{m},j}(t) < 1$, $S_j(t) \in \{0, 1\}$, and

$$|R_j(t)| \le |R_j(t-1)| + |I(t)| + V_{\mathrm{th}}. \tag{30}$$

Hence, finite first moments of the initial residual and the input current imply finite first moments of $R_j(t)$ by induction.

For the matched comparison, the two neurons receive the same realized current sequence. Subtracting their balance equations therefore cancels the input term and gives Eq. (15). Since $V_{\mathrm{th}} > 0$, Eq. (16) is both necessary and sufficient for the finite-horizon expected firing-rate ordering. Importantly, this exact identity alone does not establish that runtime temporal adaptation produces the required decay-contribution ordering.

A.2.2 Temporal Contraction and the Mean Decay Coefficient

Recall the decay coefficient

$$\alpha(\tau) = 1 - \exp\left(-\frac{\Delta t}{\tau}\right). \tag{31}$$

Its first derivative is

$$\alpha'(\tau) = -\frac{\Delta t}{\tau^2}\exp\left(-\frac{\Delta t}{\tau}\right) < 0, \tag{32}$$

so reducing the membrane timescale increases the temporal-decay coefficient.

The second derivative is

$$\alpha''(\tau) = \frac{\Delta t}{\tau^4}\exp\left(-\frac{\Delta t}{\tau}\right)(2\tau - \Delta t). \tag{33}$$

Hence, $\alpha(\tau)$ is convex on $[\Delta t/2, \infty)$.

To express the finite-horizon averages as ordinary expectations, let J be uniformly distributed on $\{1, \ldots, T\}$ and independent of the original input and neuronal randomness. Then

$$a_T = \mathbb{E}\left[\alpha(\tau_{\mathrm{A}}(J))\right] = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\alpha(\tau_{\mathrm{A}}(t))\right], \tag{34}$$

which agrees with the definition in Eq. (34). Moreover,

$$\mathbb{E}[\tau_{\mathrm{A}}(J)] = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\tau_{\mathrm{A}}(t)]. \tag{35}$$

If

$$\tau_{\mathrm{A}}(t) \geq \frac{\Delta t}{2} \quad \text{almost surely for every } t, \tag{36}$$

then $\tau_{\mathrm{A}}(J)$ lies entirely in the convex region of α. Under the mean-contraction condition in Eq. (17), Jensen's inequality gives

$$\begin{aligned} a_T &= \mathbb{E}\left[\alpha(\tau_{\mathrm{A}}(J))\right] \\ &\geq \alpha\left(\mathbb{E}[\tau_{\mathrm{A}}(J)]\right) \\ &\geq \alpha(\tau_0) = a_0. \end{aligned} \tag{37}$$

The first inequality follows from convexity, while the second follows from the monotonic decrease of α.

The support condition must apply to the adaptive timescale itself, rather than only to its mean. A simple sufficient implementation condition is

$$\tau_{\min} \geq \frac{\Delta t}{2}. \tag{38}$$

Alternatively, if

$$\tau_{\mathrm{A}}(t) \leq \tau_0 \tag{39}$$

almost surely for every timestep, monotonicity directly gives

$$\alpha(\tau_{\mathrm{A}}(t)) \geq \alpha(\tau_0), \tag{40}$$

and therefore $a_T \geq a_0$ without requiring convexity or Jensen's inequality.

A.2.3 Exact Moment Decomposition and Conditional Sparsity Bound

The increase of the mean decay coefficient does not by itself imply an increase in the total decay contribution, because $\mathcal{D}_T^{\mathrm{A}}$ depends jointly on the adaptive decay coefficient and the residual membrane state.

Using the random timestep J, define

$$m_{j,T} = \mathbb{E}[R_j(J-1)] = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[R_j(t-1)], \tag{41}$$

and

$$\delta_T = m_{\mathrm{A},T} - m_{0,T}. \tag{42}$$

Thus, δ_T measures the change in the mean residual operating point induced by runtime adaptation.

We further define the pooled decay–state covariance

$$c_T = \mathrm{Cov}\left(\alpha(\tau_{\mathrm{A}}(J)), R_{\mathrm{A}}(J-1)\right). \tag{43}$$

The covariance is taken over both the random timestep J and the original input/neuronal randomness. No independence between the adaptive timescale and residual state is assumed.

Using $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y] + \mathrm{Cov}(X,Y)$, the adaptive decay contribution becomes

$$\begin{aligned}\mathcal{D}_T^{\mathrm{A}} &= \mathbb{E}\left[\alpha(\tau_{\mathrm{A}}(J))R_{\mathrm{A}}(J-1)\right] \\ &= a_T m_{\mathrm{A},T} + c_T.\end{aligned} \tag{44}$$

For the static reference, $\alpha(\tau_0) = a_0$ is constant, and hence

$$\mathcal{D}_T^{0} = a_0 m_{0,T}. \tag{45}$$

Therefore,

$$\begin{aligned}\mathcal{D}_T^{\mathrm{A}} - \mathcal{D}_T^{0} &= a_T m_{\mathrm{A},T} - a_0 m_{0,T} + c_T \\ &= (a_T - a_0)m_{0,T} + a_T\delta_T + c_T.\end{aligned} \tag{46}$$

Eq. (46) separates three effects: the increase in the mean decay coefficient, the shift in the residual operating point, and the dependence between adaptive decay and the residual state.

Combining Eq. (46) with Eq. (15) gives the exact relation

$$\begin{aligned}V_{\mathrm{th}}\left(\bar{r}_T^{\mathrm{A}} - \bar{r}_T^{0}\right) = &-(a_T - a_0)m_{0,T} \\ &- a_T\delta_T - c_T + \Delta B_T.\end{aligned} \tag{47}$$

Proposition 1: Finite-horizon comparison with controlled terms. Suppose

$$a_T \geq a_0, \qquad m_{0,T} \geq 0, \tag{48}$$

and assume that

$$|\delta_T| \leq \varepsilon_T, \qquad c_T \geq -\kappa_T, \qquad \Delta B_T \leq b_T, \tag{49}$$

for specified nonnegative ε_T, κ_T, and b_T. Then

$$\boxed{\bar{r}_T^{\mathrm{A}} - \bar{r}_T^{0} \leq \frac{-(a_T - a_0)m_{0,T} + a_T\varepsilon_T + \kappa_T + b_T}{V_{\mathrm{th}}}.} \tag{50}$$

In particular,

$$(a_T - a_0)m_{0,T} \geq a_T\varepsilon_T + \kappa_T + b_T \tag{51}$$

is sufficient for

$$\bar{r}_T^{\mathrm{A}} \leq \bar{r}_T^{0}. \tag{52}$$

Proof. Since $a_T \geq 0$ and $|\delta_T| \leq \varepsilon_T$,

$$-a_T\delta_T \leq a_T\varepsilon_T. \tag{53}$$

Similarly,

$$-c_T \le \kappa_T, \qquad \Delta B_T \le b_T. \tag{54}$$

Substituting these inequalities into Eq. (47) gives Eq. (50). If Eq. (51) holds, the right-hand side is nonpositive, which proves the claimed firing-rate ordering.

If the sufficient condition holds with a positive margin $\gamma > 0$,

$$(a_T - a_0)m_{0,T} \ge a_T\varepsilon_T + \kappa_T + b_T + \gamma, \tag{55}$$

then the stronger bound

$$\bar{r}_T^{\mathrm{A}} \le \bar{r}_T^{0} - \frac{\gamma}{V_{\mathrm{th}}} \tag{56}$$

follows.

The quantities in Eq. (49) have direct interpretations. The term ε_T controls the mismatch between the adaptive and static mean residual states, κ_T bounds an adverse decay–state covariance, and b_T bounds the finite-horizon boundary effect. The condition $m_{0,T} \ge 0$ requires only a nonnegative *mean* residual state of the static reference and does not require $R_0(t) \ge 0$ at every timestep.

These conditions are additional assumptions rather than consequences of temporal contraction alone. In particular, $a_T \ge a_0$ only establishes an increase in the mean decay coefficient; the residual and covariance terms determine whether this increase produces a larger overall temporal-decay contribution.

A.2.4 Boundary Control and Stationary Specialization

Assume additionally that

$$\sup_{t\ge 1} \mathbb{E}|I(t)| \le K < \infty. \tag{57}$$

Let

$$q = \exp\left(-\frac{\Delta t}{\tau_{\max}}\right) < 1. \tag{58}$$

Since $\tau_j(t) \le \tau_{\max}$,

$$\rho_{\mathrm{m},j}(t) \le q \tag{59}$$

for both matched neurons. The residual recursion therefore gives

$$\mathbb{E}|R_j(t)| \le q\,\mathbb{E}|R_j(t-1)| + K + V_{\mathrm{th}}. \tag{60}$$

Iterating,

$$\mathbb{E}|R_j(t)| \le q^t\mathbb{E}|R_*| + \frac{K + V_{\mathrm{th}}}{1-q}(1-q^t) \le C, \tag{61}$$

where

$$C = \max\left\{\mathbb{E}|R_*|, \frac{K + V_{\mathrm{th}}}{1-q}\right\}. \tag{62}$$

Under the common-initial-state finite-horizon comparison,

$$R_{\mathrm{A}}(0) = R_0(0), \tag{63}$$

so

$$\Delta B_T = \frac{\mathbb{E}[R_0(T)] - \mathbb{E}[R_{\mathrm{A}}(T)]}{T}. \tag{64}$$

Therefore,

$$|\Delta B_T| \le \frac{2C}{T}. \tag{65}$$

Hence,

$$b_T = \frac{2C}{T} \tag{66}$$

is a valid, although potentially conservative, choice in Proposition 1. Thus the finite-horizon boundary difference is $O(T^{-1})$ under uniformly bounded first moments.

This boundedness result does not by itself imply convergence to a stationary distribution.

For a stationary specialization, replace the common-initial-state assumption by initialization of each neuron in its corresponding stationary regime under a common stationary input process. Assume that the relevant joint distributions of the adaptive decay coefficient and residual state are stationary and integrable. Then

$$\mathbb{E}[R_j(T)] = \mathbb{E}[R_j(0)], \qquad B_T^j = 0. \tag{67}$$

Dropping the finite-horizon subscripts, define

$$a = \mathbb{E}\left[\alpha(\tau_{\mathrm{A}}(t))\right], \tag{68}$$
$$m_j = \mathbb{E}\left[R_j(t-1)\right], \tag{69}$$
$$c = \mathrm{Cov}\left(\alpha(\tau_{\mathrm{A}}(t)), R_{\mathrm{A}}(t-1)\right). \tag{70}$$

The stationary firing-rate difference becomes

$$\begin{aligned} V_{\mathrm{th}}\left(\bar{r}^{\mathrm{A}} - \bar{r}^{0}\right) = &-(a-a_0)m_0 \\ &- a(m_{\mathrm{A}} - m_0) - c. \end{aligned} \tag{71}$$

Consequently, if

$$a \geq a_0, \qquad m_0 \geq 0, \qquad |m_{\mathrm{A}} - m_0| \leq \varepsilon, \qquad c \geq -\kappa, \tag{72}$$

and

$$(a-a_0)m_0 \geq a\varepsilon + \kappa, \tag{73}$$

then

$$\bar{r}^{\mathrm{A}} \leq \bar{r}^{0}. \tag{74}$$

The special case

$$m_{\mathrm{A}} = m_0 \geq 0, \qquad c \geq 0 \tag{75}$$

is sufficient whenever $a \geq a_0$. The same stationary identity can also be obtained as a limit of the finite-horizon identities when the relevant time-averaged quantities converge and the boundary difference vanishes.

A.2.5 Connection to the RATI-LIF Modulation Dynamics

After omitting neuron and layer indices, the runtime modulation introduced in Section 2.2 is

$$\tau_{\mathrm{input}}(t) = \lambda_i \tau_{\mathrm{input}}(t-1) + \eta_i I(t), \tag{76}$$
$$\tau_{\mathrm{output}}(t) = \lambda_o \tau_{\mathrm{output}}(t-1) + \eta_o S_{\mathrm{A}}(t-1), \tag{77}$$

with

$$\tau_{\mathrm{A}}(t) = \mathrm{clip}\left(\tau_0 - \tau_{\mathrm{input}}(t) - \tau_{\mathrm{output}}(t), \tau_{\min}, \tau_{\max}\right). \tag{78}$$

Suppose that the modulation states, input current, and output spikes are stationary with finite first moments, and

$$|\lambda_i| < 1, \qquad |\lambda_o| < 1. \tag{79}$$

Define

$$\mu_I = \mathbb{E}[I(t)], \qquad \bar{r}^{\mathrm{A}} = \mathbb{E}[S_{\mathrm{A}}(t)]. \tag{80}$$

Taking expectations in the modulation recursions gives

$$\mathbb{E}[\tau_{\mathrm{input}}] = \frac{\eta_i}{1-\lambda_i}\mu_I, \qquad \mathbb{E}[\tau_{\mathrm{output}}] = \frac{\eta_o}{1-\lambda_o}\bar{r}^{\mathrm{A}}. \tag{81}$$

These identities are exact under the stated stationarity assumptions. The parameter bounds on λ_i and λ_o alone do not establish stationarity of the coupled neuronal system.

Define the unclipped runtime timescale by

$$\widetilde{\tau}(t) = \tau_0 - \tau_{\mathrm{input}}(t) - \tau_{\mathrm{output}}(t), \tag{82}$$

and let

$$(x)_+ = \max\{x, 0\}. \tag{83}$$

For any $a \le b$,

$$\mathrm{clip}(x, a, b) = x + (a - x)_+ - (x - b)_+. \tag{84}$$

Therefore,

$$\begin{aligned}\mathbb{E}[\tau_{\mathrm{A}}] = \tau_0 - \frac{\eta_i}{1-\lambda_i}\mu_I - \frac{\eta_o}{1-\lambda_o}\bar{r}^{\mathrm{A}} \\ + \mathbb{E}\left[(\tau_{\min} - \widetilde{\tau})_+\right] - \mathbb{E}\left[(\widetilde{\tau} - \tau_{\max})_+\right].\end{aligned} \tag{85}$$

Consequently,

$$\mathbb{E}[\tau_{\mathrm{A}}] \le \tau_0 \tag{86}$$

holds if and only if

$$\begin{aligned}\frac{\eta_i}{1-\lambda_i}\mu_I + \frac{\eta_o}{1-\lambda_o}\bar{r}^{\mathrm{A}} \ge \mathbb{E}\left[(\tau_{\min} - \widetilde{\tau})_+\right] \\ - \mathbb{E}\left[(\widetilde{\tau} - \tau_{\max})_+\right].\end{aligned} \tag{87}$$

If clipping is inactive almost surely, both correction terms vanish. When clipping is active, positive mean net modulation alone does not generally guarantee mean timescale contraction.

A separate sufficient condition is

$$\tau_{\mathrm{input}}(t) + \tau_{\mathrm{output}}(t) \ge 0 \tag{88}$$

almost surely for every timestep. Since the clipping operator is monotone and $\tau_0 \in [\tau_{\min}, \tau_{\max}]$,

$$\tau_{\mathrm{A}}(t) \le \mathrm{clip}\left(\tau_0, \tau_{\min}, \tau_{\max}\right) = \tau_0. \tag{89}$$

This pointwise conclusion requires neither stationarity nor an approximation that ignores clipping.

A.2.6 A Stronger Result under Nonnegative Residuals

Proposition 2: Pathwise cumulative spike-count ordering. Consider the matched binary-spike, subtractive-reset dynamics. Suppose that, for each input realization,

$$R_{\mathrm{A}}(0) = R_0(0), \qquad R_{\mathrm{A}}(t-1) \ge 0, \qquad 0 \le \rho_{\mathrm{m},A}(t) \le \rho_{\mathrm{m},0} \le 1 \tag{90}$$

for all relevant timesteps.

Define the cumulative spike count

$$N_j(t) = \sum_{s=1}^{t} S_j(s), \qquad N_j(0) = 0. \tag{91}$$

Then

$$\boxed{N_{\mathrm{A}}(T) \le N_0(T) \qquad \text{for every finite } T.} \tag{92}$$

Proof. Define

$$Q_j(t) = R_j(t) + V_{\mathrm{th}} N_j(t), \tag{93}$$

together with

$$\Delta N(t) = N_0(t) - N_{\mathrm{A}}(t), \qquad \Delta Q(t) = Q_0(t) - Q_{\mathrm{A}}(t). \tag{94}$$

We prove by induction that

$$\Delta N(t) \ge 0, \qquad \Delta Q(t) \ge 0. \tag{95}$$

Both are zero at $t = 0$.

Suppose the inequalities hold at $t-1$. Using the residual dynamics and $N_j(t) = N_j(t-1) + S_j(t)$, the subtractive-reset term cancels:

$$Q_j(t) = \rho_{\mathrm{m},j}(t) R_j(t-1) + I(t) + V_{\mathrm{th}} N_j(t-1). \tag{96}$$

Therefore,

$$\begin{aligned}\Delta Q(t) = \rho_{\mathrm{m},0} \Delta Q(t-1) + V_{\mathrm{th}}\left(1 - \rho_{\mathrm{m},0}\right)\Delta N(t-1) \\ + \left(\rho_{\mathrm{m},0} - \rho_{\mathrm{m},A}(t)\right) R_{\mathrm{A}}(t-1) \ge 0.\end{aligned} \tag{97}$$

If $\Delta N(t-1) \geq 1$, then, because each neuron emits at most one spike per timestep,

$$\Delta N(t) = \Delta N(t-1) + S_0(t) - S_{\mathrm{A}}(t) \geq 0. \tag{98}$$

If instead $\Delta N(t-1) = 0$, then

$$R_0(t-1) - R_{\mathrm{A}}(t-1) = \Delta Q(t-1) \geq 0. \tag{99}$$

Hence,

$$\begin{aligned} H_0(t) - H_{\mathrm{A}}(t) = \rho_{\mathrm{m},0}\left(R_0(t-1) - R_{\mathrm{A}}(t-1)\right) \\ + \left(\rho_{\mathrm{m},0} - \rho_{\mathrm{m},A}(t)\right) R_{\mathrm{A}}(t-1) \geq 0. \end{aligned} \tag{100}$$

Since the two neurons share the same threshold,

$$S_0(t) \geq S_{\mathrm{A}}(t), \tag{101}$$

which again gives $\Delta N(t) \geq 0$. This completes the induction.

If Eq. (90) holds almost surely, then taking expectations in Eq. (92) yields

$$\bar{r}_T^{\mathrm{A}} \leq \bar{r}_T^0 \tag{102}$$

for every finite T.

The result concerns cumulative spike counts; it does not require $S_{\mathrm{A}}(t) \leq S_0(t)$ at every individual timestep.

Corollary: RATI-LIF under nonnegative inputs. Assume the common initial residual satisfies

$$R_* \geq 0, \tag{103}$$

and

$$\begin{gathered} I(t) \geq 0, \qquad \tau_{\mathrm{input}}(0) \geq 0, \qquad \tau_{\mathrm{output}}(0) \geq 0, \\ 0 \leq \lambda_i, \lambda_o < 1, \qquad \eta_i, \eta_o \geq 0 \end{gathered} \tag{104}$$

almost surely.

The modulation recursions then preserve

$$\tau_{\mathrm{input}}(t) \geq 0, \qquad \tau_{\mathrm{output}}(t) \geq 0. \tag{105}$$

Eq. (89) implies

$$\tau_{\mathrm{A}}(t) \leq \tau_0, \tag{106}$$

and therefore

$$\rho_{\mathrm{m},A}(t) \leq \rho_{\mathrm{m},0}. \tag{107}$$

It remains to show that the adaptive residual remains nonnegative. If $R_{\mathrm{A}}(t-1) \geq 0$, then

$$H_{\mathrm{A}}(t) = \rho_{\mathrm{m},A}(t) R_{\mathrm{A}}(t-1) + I(t) \geq 0. \tag{108}$$

If the neuron does not spike,

$$R_{\mathrm{A}}(t) = H_{\mathrm{A}}(t) \geq 0. \tag{109}$$

If it spikes, then $H_{\mathrm{A}}(t) \geq V_{\mathrm{th}}$, giving

$$R_{\mathrm{A}}(t) = H_{\mathrm{A}}(t) - V_{\mathrm{th}} \geq 0. \tag{110}$$

Induction from $R_* \geq 0$ therefore establishes $R_{\mathrm{A}}(t) \geq 0$ for every timestep. Proposition 2 then gives

$$N_{\mathrm{A}}(T) \leq N_0(T) \tag{111}$$

almost surely for every finite T, and consequently

$$\bar{r}_T^{\mathrm{A}} \leq \bar{r}_T^0. \tag{112}$$

This corollary requires neither stationarity nor assumptions on residual-mean equality or decay–state covariance. It also does not require $\tau_{\min} \geq \Delta t/2$, because it relies on pointwise monotonicity rather than Jensen's inequality. The guarantee is a non-increase in firing rather than necessarily a strict reduction.

Nonnegative input is an additional model restriction. Binary presynaptic spikes do not imply nonnegative postsynaptic currents when synaptic weights or biases may be negative. For signed inputs, Proposition 1 remains applicable, whereas this corollary requires its stronger nonnegativity assumptions.

A.2.7 Scope of the Matched Comparison

For signed residual states, reducing the retention factor can increase the membrane potential by attenuating a negative residual. Specifically, if

$$R_{\mathrm{A}}(t-1) < 0, \tag{113}$$

then decreasing $\rho_{\mathrm{m},A}(t)$ makes $\rho_{\mathrm{m},A}(t)R_{\mathrm{A}}(t-1)$ less negative. Thus, temporal contraction alone does not guarantee reduced firing in the general signed-input setting.

The matched comparison requires the same realized current sequence at the neurons being compared. In a multilayer network, changing upstream neuronal dynamics can alter upstream spike trains and therefore the input currents received by downstream neurons even when the synaptic weights are fixed. A direct neuronal comparison can instead be implemented by replaying the same recorded current sequence.

For two networks A and B with the same firing threshold and subtractive reset rule but potentially different input currents, the same balance argument gives

$$\begin{aligned} V_{\mathrm{th}}\left(\bar{r}_T^A - \bar{r}_T^B\right) = &\left(\mu_{I,T}^A - \mu_{I,T}^B\right) \\ &- \left(\mathcal{D}_T^A - \mathcal{D}_T^B\right) + \left(B_T^A - B_T^B\right). \end{aligned} \tag{114}$$

Hence, changes in the input drive may either offset or reinforce changes in the temporal-decay contribution.

The theoretical result therefore establishes explicit conditions for reduced firing under a matched neuronal comparison. It does not imply an unconditional firing-rate ordering between independently optimized LIF, PLIF, and RATI-LIF networks.

A.3 Detailed Description of the RATI-LIF Neuron

The RATI-LIF neuron extends the standard LIF model by introducing an adaptive membrane time constant that is modulated intrinsically by neuronal signals. At each time step, two auxiliary state variables are maintained to capture input-driven and spike-driven modulation, respectively. These variables are updated using first-order recursions and combined with a learnable base membrane time constant to compute the adaptive time constant, which is clipped to a predefined range to ensure numerical stability. The membrane decay factor is then recomputed at each time step based on the adaptive time constant, while the membrane potential update and spike generation follow the standard LIF formulation. Importantly, the RATI-LIF neuron introduces no additional network branches and only a constant number of extra state variables, preserving computational efficiency. The complete update procedure of the RATI-LIF neuron is summarized in Algorithm 1.

Algorithm 1 RATI-LIF Neuron Update (Single Time Step)

Require: Input current $I(t)$, previous membrane potential $H(t-1)$, previous spike $S(t-1)$
Require: Previous modulation states $\tau_{\text{input}}(t-1)$, $\tau_{\text{output}}(t-1)$
Require: Learnable base time constant τ_0
Require: Parameters $\lambda_i, \lambda_o, \eta_i, \eta_o$, bounds $\tau_{\min}, \tau_{\max}$
$\tau_{\text{input}}(t) \leftarrow \lambda_i \tau_{\text{input}}(t-1) + \eta_i I(t)$
$\tau_{\text{output}}(t) \leftarrow \lambda_o \tau_{\text{output}}(t-1) + \eta_o S(t-1)$
$\tau(t) \leftarrow \mathrm{clip}\big(\tau_0 - \tau_{\text{input}}(t) - \tau_{\text{output}}(t), \tau_{\min}, \tau_{\max}\big)$
$\rho_m(t) \leftarrow \exp\left(-\frac{\Delta t}{\tau(t)}\right)$
$U(t) \leftarrow \rho_m(t)\left(H(t-1) - S(t-1)V_{\mathrm{th}}\right)$
$H(t) \leftarrow U(t) + I(t)$
$S(t) \leftarrow \mathbb{I}\big(H(t) \geq V_{\mathrm{th}}\big)$
Return $H(t), S(t), \tau_{\text{input}}(t), \tau_{\text{output}}(t)$

A.4 Experimental Details

A.4.1 Datasets

In this paper, we perform experiments on the EEG dataset DEAP, the neuromorphic vision datasets CIFAR10-DVS and DVS-Gesture, and the event-based speech datasets SHD and SSC.

DEAP (Koelstra et al., 2011): A multi-modal dataset for the analysis of human affective states. It contains electroencephalography and other physiological recordings from 32 participants, each watching 40 one-minute-long music videos. The dataset is labeled with ratings for arousal, valence, dominance, and liking, making it a standard benchmark for emotion recognition tasks using physiological signals.

DVS-Gesture (Amir et al., 2017): A neuromorphic benchmark dataset for gesture recognition, recorded using a DVS128 dynamic vision sensor. It comprises 1,342 instances of 11 hand and arm gestures, organized into 122 trials collected from 29 subjects under three different lighting conditions. During each trial, one subject sequentially performs all 11 gestures under the same lighting condition. Following the preprocessing adopted by SpikingJelly, the segmented dataset used in our experiments contains 1,176 training samples and 288 test samples. The event-based recordings provide rich temporal information, making DVS-Gesture a widely used benchmark for evaluating temporal processing in spiking neural networks.

CIFAR10-DVS (Li et al., 2017): A neuromorphic version of the popular CIFAR-10 dataset, converted by displaying the original images on a monitor and recording the output with a dynamic vision sensor. It contains 10,000 event streams spanning 10 object classes, presenting a challenging benchmark for object recognition with event-based cameras that requires handling sparse, asynchronous inputs.

SHD (Cramer et al., 2020): The Spiking Heidelberg Digits dataset is a spike-based speech recognition benchmark consisting of spoken digits from zero to nine in both English and German, resulting in 20 classes. The original audio recordings are converted into spike trains over 700 input channels using a biologically inspired artificial cochlea model. The dataset contains 8,156 training samples and 2,264 test samples, providing a benchmark for evaluating temporal processing in SNNs.

SSC (Cramer et al., 2020): The Spiking Speech Commands dataset is a larger-scale spike-based speech benchmark derived from the Google Speech Commands v0.2 dataset. It contains 35 spoken-word classes with 75,466 training samples, 9,981 validation samples, and 20,382 test samples. Similar to SHD, the audio waveforms are converted into spike trains over 700 input channels using an artificial cochlea model, making SSC suitable for evaluating SNNs on large-scale and temporally complex speech sequences.

A.4.2 Training Setup

Experiments are implemented using PyTorch (version 2.0.1) with CUDA 11.8 and conducted on a server equipped with NVIDIA RTX 4090 and NVIDIA RTX PRO 6000 GPUs. In the experiment section, the details of the simulation time steps, training epochs, batch size, and some neuronal hyperparameters are summarized in Tab. 5.

For the DEAP dataset, we adopt a two-layer fully connected network structure with dimensions [512, 3/2], while for the SHD and SSC datasets, we employ single- and two-hidden-layer fully connected networks with different hidden dimensions, as detailed in the corresponding speech experiment table. The detailed hyperparameter settings are provided in Tab. 5. For the neuromorphic vision datasets DVS-Gesture and CIFAR10-DVS, as experiments were conducted across different network architectures, parameters such as the number of time steps and the firing threshold are not fixed, but instead follow the configurations used in previous works (Meng et al., 2023; Zhou et al., 2024; Zheng et al., 2021; Fang et al., 2021a; Zhou et al., 2022). The only modification in these experiments is the replacement of the original LIF or PLIF neurons in the reference architectures with our proposed RATI-LIF neurons for a fair comparison.

Table 5: Parameter settings on different learning tasks.

Parameter	Description	DEAP	DVS-Gesture	CIFAR10-DVS	SHD	SSC
Batch size	–	200	–	–	256	128
T	Time steps	6	–	–	100	100
λ_i, λ_o	Decay factors	0.1	0.35	0.2	0.1	0.1
η_i, η_o	Activity impact factors	0.01	0.1	0.05	0.1	0.1
V_{th}	Threshold	1.0	–	–	1.0	1.0
N	Training epochs	100	–	–	150	100

A.4.3 Neuronal Model Variants

Specifically, we subject the input events to Gaussian noise and salt-and-pepper noise, and compare the performance across four neuronal variants: standard LIF, LIF with learnable τ, LIF with dynamic τ, and our proposed RATI-LIF. The standard LIF neuron follows the formulation adopted in Zheng et al. (2024).

As demonstrated by Eq. (4), the non-differentiable nature of spiking activity precludes the direct application of backpropagation to SNNs. To address this fundamental limitation, surrogate gradient methods replace the undefined gradients of spike generation with a differentiable surrogate function $u(\cdot)$ during error backpropagation. Specifically, while the spike activity is computed according to Eq. (4), its derivative with respect to the membrane potential is approximated as

$$\frac{\partial S^l(t)}{\partial U^l(t)} \approx u'\left(U^l(t), V_{\text{th}}\right), \tag{115}$$

which enables gradient-based optimization. We follow prior works (Wu et al., 2018; 2022; Ding et al., 2024b) and adopt a rectangular surrogate gradient function defined as

$$u'\left(U^l(t), V_{\text{th}}\right) = \frac{1}{a}\,\text{sign}\Big(\left|U^l(t) - V_{\text{th}}\right| < \frac{a}{2}\Big), \tag{116}$$

where a is a hyperparameter controlling the width of the rectangular window. Under this surrogate gradient framework, model parameters that influence the membrane potential dynamics can be optimized via standard backpropagation. In particular, for the LIF with learnable τ variant, the membrane time constant is treated as a trainable parameter and updated jointly with synaptic weights through the same surrogate gradient mechanism during training, while remaining fixed during inference.

For the LIF with dynamic τ variant, we adopt the same dynamic time-constant formulation as described in Section 2.2. Specifically, the membrane timescale is modulated by both input intensity and recent spiking activity through input-driven and spike-driven components, while the baseline time constant τ_0 is fixed and non-learnable in this variant. All other settings follow the formulation introduced in the main text.

A.5 Additional Ablation on DVS-Gesture

To examine whether the contributions of the learnable reference timescale and runtime adaptation observed on DEAP generalize to a different temporal modality, we conduct the same ablation study on DVS-Gesture. We compare the fixed-timescale LIF baseline, a learnable reference timescale without runtime adaptation, runtime adaptation with a fixed reference timescale, and the full RATI-LIF neuron.

As shown in Table 6, learning a static reference timescale improves accuracy from 87.50% to 95.14%, while introducing runtime adaptation with a fixed reference timescale reaches 95.49%. Combining both components in RATI-LIF further improves accuracy to 96.18%. This result is consistent with the DEAP ablation reported in the main text, providing additional evidence that the learnable reference timescale and runtime adaptation make complementary contributions across different temporal data modalities.

Table 6: Additional ablation study of the learnable reference timescale and runtime adaptation on DVS-Gesture. Values in parentheses indicate accuracy improvements (in percentage points) over the LIF neuron model.

Model	Accuracy (%)
LIF	87.50
+ learnable reference τ_0	95.14 (+7.64)
+ runtime-adaptive $\tau(t)$	95.49 (+7.99)
RATI-LIF	**96.18 (+8.68)**

A.6 Ablation Study on λ and η Coefficients

The selection of hyperparameters plays a critical role in overall network performance. To evaluate the impact of the key hyperparameters λ and η introduced in our RATI-LIF model, we conduct a hyperparameter search on the DEAP dataset (three-class valence emotion classification task). We begin with a coarse-grained evaluation by setting both λ and η to values in $\{0.001, 0.01, 0.1, 0.2\}$. The corresponding accuracies—80.89%, 88.55%, 86.33%, and 79.88%—clearly indicate that very small (e.g., 0.001) or very large (e.g., 0.2) values significantly degrade performance, underscoring the importance of appropriate hyperparameter balancing. Based on these observations, we perform a finer-grained search within λ in $\{0.01, 0.05, 0.1, 0.15\}$ and η in $\{0.005, 0.01, 0.05, 0.1, 0.15\}$. As summarized in Tab. 7, the optimal performance is achieved at λ=0.1 and η=0.01, yielding 90.11% accuracy on the DEAP dataset. These results confirm that identifying suitable values for λ and η leads to substantially improved performance of the proposed neuron model.

Table 7: Accuracy under different settings.

λ \ η	0.005	0.01	0.05	0.1	0.15
0.01	88.71	88.55	86.72	87.65	86.52
0.05	87.81	88.98	88.55	87.34	87.77
0.1	88.77	**90.11**	87.46	86.33	86.41
0.15	87.34	88.86	87.42	85.62	83.71

A.7 Robustness under Input Perturbations

Although robustness is not the primary focus of this work, runtime adaptation may also help neurons respond to changes in input conditions. We therefore provide an additional evaluation under Gaussian and salt-and-pepper perturbations on DVS-Gesture. We compare standard LIF, the learnable-reference variant, the runtime-adaptive-only variant, and the full RATI-LIF neuron. Detailed definitions of the compared neuronal variants are provided in Appendix A.4.3.

A.7.1 Robustness Experiments

We evaluate the robustness of our model under input perturbations. This is critical for neuromorphic systems that operate in noisy environments, such as low-light settings or high-speed motion scenarios (Hendy & Merkel, 2022). We apply two widely used noise types: Gaussian noise and salt-and-pepper noise, both commonly used in robustness testing of SNNs (Li et al., 2020; Wu et al., 2022; Ding et al., 2024a).

Gaussian noise: We perturb the input by adding zero-mean Gaussian noise sampled from $\mathcal{N}(0, \sigma^2)$ to each pixel of the image. The standard deviation is controlled via the variance parameter σ^2. We define ten increasing noise levels as follows:

$$\sigma^2 \in \{0.01,\ 0.02,\ 0.03,\ 0.04,\ 0.05,\ 0.06,\ 0.07,\ 0.08,\ 0.09,\ 0.10\}.$$

Each noise level is applied independently to the test set, and model performance is reported as classification accuracy. This simulates gradually deteriorating sensor quality or transmission interference.

Salt-and-pepper noise: This noise simulates dead pixels and random corruption by flipping a proportion of image pixels to 0 (black) or 1 (white). For each level, we randomly select a fraction a of pixels in the image and assign them new values drawn from a binary distribution. The tested levels are:

$$a \in \{0.02,\ 0.04,\ 0.06,\ 0.08,\ 0.10,\ 0.12,\ 0.14,\ 0.16,\ 0.18,\ 0.20\}.$$

This experiment tests the model's ability to recognize structure in partially corrupted input and maintain stability under non-Gaussian perturbations.

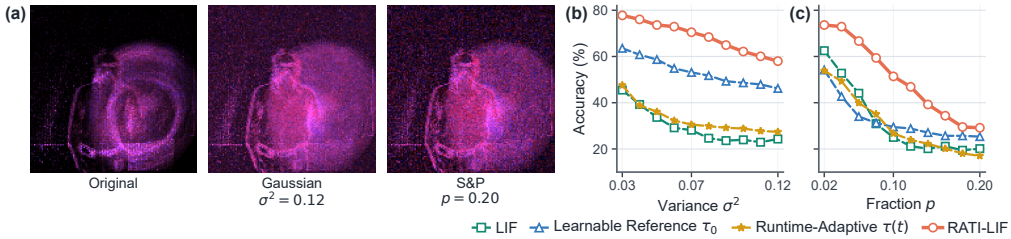

Figure 3: Robustness evaluation on DVS-Gesture under Gaussian and salt-and-pepper perturbations. (a) Representative input samples under the original condition, Gaussian noise with variance $\sigma^2 = 0.12$, and salt-and-pepper noise with corruption fraction $p = 0.20$. (b) Classification accuracy under increasing Gaussian noise variance σ^2. (c) Classification accuracy under increasing salt-and-pepper corruption fraction p. We compare standard LIF, the learnable reference τ_0, the runtime-adaptive $\tau(t)$ variant, and the full RATI-LIF neuron.

As shown in Fig. 3, increasing either Gaussian or salt-and-pepper corruption progressively degrades classification performance across all neuronal variants. RATI-LIF consistently maintains stronger performance than standard LIF, the learnable-reference variant, and the runtime-adaptive-only variant across the evaluated perturbation levels. These results provide complementary evidence that combining a task-dependent reference timescale with runtime adaptation can improve stability under changing input conditions, while robustness remains a secondary observation relative to the primary gains in firing sparsity, classification accuracy, and estimated energy efficiency.