ROBUST RESONATE-AND-FIRE NEURONS FOR REAL-TIME END-TO-END EEG EMOTION RECOGNITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world applications of EEG analysis call for energy-efficient, low latency operation to enable scalable, real-time inference on streamed signals. Spiking Neural Networks (SNNs), through sparse, event-driven processing, can satisfy these demands, yet their benefits are rarely realised in practice as implementations stick to pipelines that feature dense, windowed computation. This includes SNNs for EEG emotion recognition, where the use of windowed spectral features hampers sparse computation, inflates latency, and coarsens temporal resolution. More elaborate neuron models can address these limitations: the Resonate-and-Fire (RF) model captures spectral features via oscillating membrane dynamics, but remains under-explored due to parameter sensitivity and instability. Existing RF formulations mitigate these issues at the cost of added complexity; we instead introduce a robust RF model, bolstered by data-driven initialisation, that enables reliable and efficient integration into SNNs, as evidenced by reduced computational cost and spike rates compared to other RF models. Leveraging our RF neurons to extract spectral features directly from raw EEG, we further present **SS-EEGNet**, a Spiking and Streaming EEGNet-based model for online end-to-end emotion recognition in real-time. Our approach enables causal, low-latency inference while preserving sparsity, achieving competitive accuracy alongside substantial efficiency gains.

1 INTRODUCTION

While deep neural networks have achieved strong results in Electroencephalography (EEG)-based Emotion Recognition (ER), their heavy computational demands make them poorly suited for real-world applications in real-time and resource-constrained settings. Accordingly, there is increasing demand for edge computing approaches that employ lightweight models closer to the data source, reducing latency, bandwidth, and energy use, along with improving privacy. To this end, we turn to *neuromorphic computing*, which offers highly efficient methods and hardware by mimicking biological processes, making real-world deployment of deep learning algorithms possible.

These efficiency gains are primarily driven by emulating the spiking dynamics of biological neurons in Spiking Neural Networks (SNNs), which typically employ (variants of) the *Leaky Integrate-and-Fire* (LIF) neuron (Eshraghian et al., 2023). It is this event-driven sparsity – spiking neurons remain silent until they fire after *integrating* enough input – that underpins the remarkable efficiency of neuromorphic systems. SNNs have already demonstrated substantial efficiency gains across a variety of tasks, but generally sacrifice accuracy to do so (Kudithipudi et al., 2025). Furthermore, most state-of-the-art (SOTA) SNNs for EEG ER still rely on non-spiking, windowed features, such as the Short-Time Fourier Transform (STFT) or temporal convolutions (Xu et al., 2024; Zhao et al., 2025); their need to aggregate windows of data makes these approaches unfit for real-time stream processing.

Resonate-and-Fire (RF) neurons offer a well-founded way to eliminate this reliance on non-spiking features by extracting spectral features directly from the input stream, enabling end-to-end EEG ER. Rooted in biological neural dynamics, the RF model extends LIF neurons with a complex-valued membrane potential that produces damped sub-threshold oscillations (Izhikevich, 2001), inducing a preference for inputs near their intrinsic resonant frequency rather than responding monotonically to input rate (Figure 1).

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.

While these dynamics can efficiently encode spectral information into spikes (Orchard et al., 2021; Hille et al., 2022; Lehmann et al., 2023), they also introduce second-order interactions between frequency, damping, and threshold that are absent in LIF models – leading to ad hoc parametrisation, sensitivity to initialisation, and unstable firing behaviour (Orchard et al., 2021; Higuchi et al., 2024). Addressing these limitations is essential for reliable use of RF neurons in SNNs, including the end-to-end EEG ER approach proposed here.

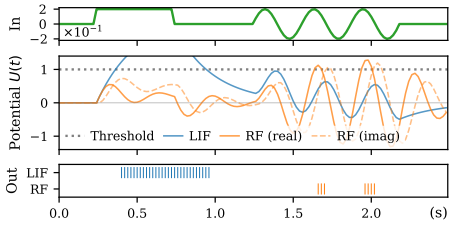


Figure 1: LIF vs. RF dynamics.

To sum up our contributions, we propose:

- **A robust RF model**, reformulated to sustain its simplicity – both computationally and conceptually – implemented in *snnTorch*[1]. Combined with practical initialisation strategies, this RF model significantly reduces sensitivity to parametrisation, reduces energy use, and enables reliable integration into SNNs.
- **SS-EEGNet**: a *Spiking Streaming* adaptation of EEGNet for streaming end-to-end EEG ER in real-time. By leveraging RF neurons for spectral analysis, we achieve efficient and continuous inference along with SOTA performance on the DEAP and SEED-VII datasets.

2 Background

2.1 Resonate-and-Fire Neurons

The RF neuron model was introduced by (Izhikevich, 2001) as an oscillating analogue of the LIF model through the single addition of an imaginary rotation factor ω. The resulting 2nd-order (complex-valued) dynamics give rise to distinct properties – most notably rhythmic firing and selectivity to specific input frequencies – which have since been leveraged in radar signal processing (Hille et al., 2022), hand gesture recognition (Shaaban et al., 2024), optical flow estimation, and audio classification (Orchard et al., 2021). However, these implementations employ ad hoc parametrisations that neglect the non-trivial interactions between RF parameters, leading to widely varying reported values, high sensitivity to initialisation, and unstable training dynamics. While recent variants address these issues via architectural complications (Higuchi et al., 2024; Zhang et al., 2025), we instead argue that robust RF behaviour can be achieved with the model's intended simplicity intact, focusing on stable parametrisation and intuitive initialisation strategies, as detailed in Section 3.

2.2 SNNs and End-to-end Approaches for EEG ER

Recent SNN-based EEG ER approaches such as (Xu et al., 2024; Zhao et al., 2025) have managed to match SOTA accuracy with improvements to efficiency, yet remain reliant on dense windowed features (e.g., STFT) or temporal convolutions/pooling, which are fundamentally incompatible with real-time stream processing. End-to-end approaches – commonly based on general architectures like EEGNet (Lawhern et al., 2018) – have demonstrated that deep networks can perform EEG ER without requiring these handcrafted features (Wang et al., 2018; Dzieżyc et al., 2020; Han et al., 2022), but being non-spiking limits their efficiency. Few attempts have been made at combining both paradigms, but these tend to still rely on temporal aggregation at key stages: (Gong et al., 2023) processes EEG sequentially through a graph-convolutional core yet in large steps of 0.5s, while (Li et al., 2024) brackets a true stepwise spiking core with encoding/decoding stages that require the full signal. In contrast, our approach functions in real-time *without any* temporal aggregation and instead relies on the recurrent characteristics inherent of spiking neurons to enable real-time stream processing.

[1]Link to GitHub repository to be included here.

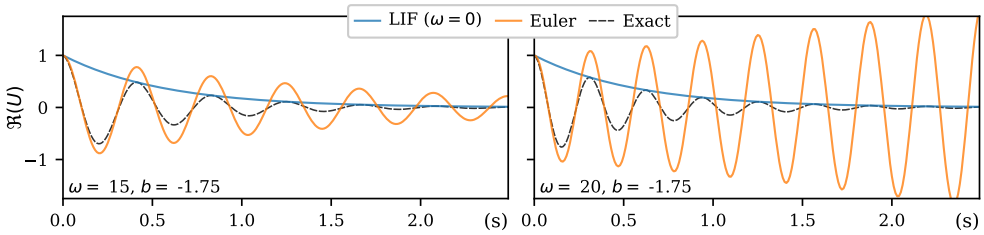


Figure 2: Comparing neuron dynamics when simulated via the forward Euler and the exact method. The oscillatory dynamics of RF neurons result in amplitude errors and instability for Euler's method.

3 Robust Resonate-and-Fire Neurons

The core subthreshold dynamics of RF neurons are defined by the linear system

$$\frac{d}{dt}z(t) = (b + i\omega)z(t) + I(t), \tag{1}$$

with $z = (x + iy) \in \mathbb{C}$, $\omega > 0$ as the angular frequency, and $b < 0$ the damping factor. This continuous-time model requires discretisation for use in SNNs; oscillatory dynamics increase the demand for precision to remain stable (Figure 2), for which we apply exact integration (Rotter & Diesmann, 1999) (full derivation in Section A). Incorporating sequential time $t_n \to n\Delta t$ and a weighted input $W \cdot X$ gives the base discretised RF update rule:

$$\beta = e^{b\Delta t}\,\text{(decay)}, \quad \rho = e^{i\omega\Delta t}\text{(rotation)}, \quad \gamma = \beta \cdot \rho,$$
$$U[t+1] = \gamma \cdot U[t] + W \cdot X[t+1]\,, \tag{RF}$$

with $\gamma, U \in \mathbb{C}$. Spiking $S[t]$ follows from the Heaviside function Θ given U_{spk} and the threshold ϑ:

$$S[t] = \Theta(U_{\text{spk}}[t] - \vartheta), \qquad U_{\text{spk}}[t] \in \{\Re(U[t]), \Im(U[t])\}\,. \tag{2}$$

From here, our robust RF model introduces 1) the derived threshold, 2) sparsity mechanisms, and 3) improved initialisation.

3.1 Threshold Scaling

Appropriately setting the threshold is key for attaining informative and sparse firing behaviour. Defaulting to $\vartheta = 1$ and relying on learning W to scale the input is effective for LIF neurons, but proves insufficient for RF neurons, as W will end up dominating the oscillatory dynamics. To avoid introducing additional scaling terms into the update rule, we instead alter the threshold to regulate spiking behaviour, as demonstrated in Figure 3. We base this threshold on the maximum achievable amplitude of the RF neuron when driven by an optimal input current (a resonant sinusoid). Following the derivation provided in Section B, the scaled threshold is:

$$\vartheta = \varepsilon \cdot (1 - \beta^T) \cdot \underbrace{\left(\frac{\sqrt{b^2 + \omega^2}}{\sqrt{b^2(b^2 + 4\omega^2)}}\right)}_{\text{max. driven amplitude}}, \tag{3}$$

with $b = \ln(\beta)$, $\varepsilon > 0$ as a simple scaling factor (analogous to the expected input amplitude), and $T > 0$ as the number of time steps over which resonant input is expected to persist. Although this introduces two additional hyperparameters, the underlying, streamlined state dynamics remain unaffected, and moreover, T and ε provide more interpretable means of regulating spiking behaviour, which we find empirically as well.

3.2 Reset & Refractory Mechanisms

To promote sparsity, we implement *both* reset and refractory mechanisms. While reset mechanisms already serve as efficient abstractions of (biological) refractory periods, explicit refractory mechanisms suppress spiking without directly altering the membrane potential, thereby better preserving

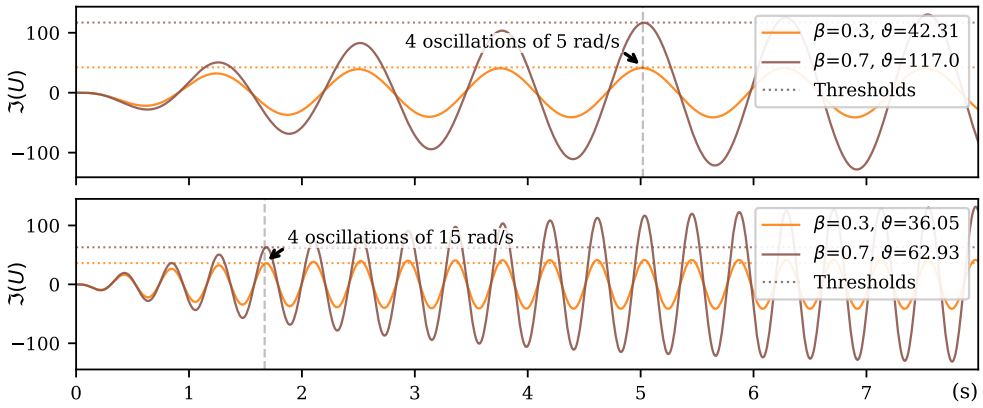


Figure 3: Demonstration of ϑ scaling across different RF neurons. Each neuron was driven by a resonant input with $A = 1$, while ϑ was derived from Equation (3) with $\varepsilon = 1$ and $T = 4 \cdot 2\pi/\omega$.

oscillatory behaviour. A complete showcase of both sparsity mechanisms is provided in Section C. Formally, resets $R(\cdot)$ are applied *after* the update rule, based on spiking in the previous step:

$$\begin{aligned} \tilde{U}[t+1] &= \gamma \cdot U[t] + W \cdot X[t+1] \\ U[t+1] &= R\left(\tilde{U}[t+1],\ S[t]\right) \end{aligned} \tag{RF$_{\text{reset}}$}$$

Given the complex arithmetic of the RF model, $R(\cdot)$ can affect the real and imaginary parts of the membrane potential independently for a wider array of options (compared to LIF neurons). Here, we highlight a handful of mechanisms (for spiking on $\Im$) we found most effective:

$$R(U,S) = \begin{cases} U, & \text{no reset [default],} \\ U - U \cdot q[t] \cdot \Delta t, & \text{smooth reset (Higuchi et al., 2024),} \\ U - S \cdot \operatorname{sgn}(U) \cdot \vartheta \cdot \beta, & \text{soft reset,} \\ U - S \cdot U, & \text{hard reset,} \end{cases} \tag{4}$$

with $\operatorname{sgn}(x) = x/|x|$ as the sign of a complex number, and $q[t]$ as a time-varying decay factor. Refractory mechanisms are implemented by modulating the base threshold ϑ' via $r[t]$ as follows:

$$\vartheta[t] = \begin{cases} \vartheta', & & \text{no ref. [default],} \\ \vartheta' + \infty \cdot \mathbf{1}_{r[t]>0}; & r[t+1] = \max(r[t]-1, 0) + S[t] \cdot \lceil \frac{\tau_{\text{ref}} 2\pi}{\omega} \rceil & \text{absolute ref.,} \\ \vartheta' + \infty \cdot \mathbf{1}_{r[t]>0}; & r[t+1] = \mathbf{1}_{U[t] \geq \vartheta'} \cdot (r[t] + S[t]) & \text{conditional ref.,} \\ \vartheta' + r[t]; & r[t+1] = r[t] \cdot \tau_{\text{ref}} + S[t] \cdot \vartheta' & \text{adaptive ref.} \end{cases} \tag{5}$$

3.3 Continuous Time Initialisation

RF neurons initialise simply from per-step values of β and ω via Algorithm 1. For efficiency, γ is fixed coefficient during forward passes, updated only after backpropagation.

To achieve consistent oscillations with respect to continuous time – e.g. when working with real-world data and frequency-based values – β and ω must be rescaled *together* from per-second to per-step values. Given a continuous-time angular frequency ω_{cont} in rad/s and a decay factor $\beta_{\text{cont}} \in [0, 1]$ as the fraction of membrane potential retained after one second, the corresponding per-step parameters are

$$\omega = \omega_{\text{cont}} \cdot \Delta t, \qquad \beta = (\beta_{\text{cont}})^{\Delta t}. \tag{6}$$

This rescaling is automatically handled in Algorithm 1, but note the different time frames of the trainable parameters: $\omega_{\text{step}} \in [0, \pi]$ and $\beta_{\text{cont}} \in [0, 1]$ are both well-conditioned ranges that support stable training, whereas β_{step} quickly approaches 1 for $\Delta t \ll 1$.

Algorithm 1 RF Initialisation

Require: β, ω
Optional: $\vartheta = 1, \Delta t = 1, \varepsilon = \text{None}, T = \text{None}, \ldots$
  class Resonate:
  if ε is not None or T is not None **then**
    $\varepsilon \leftarrow 1$ if $\varepsilon = \text{None}$ else ε
    $T \leftarrow \infty$ if $T = \text{None}$ else T
    $\vartheta \leftarrow \text{derive_threshold}(\beta, \omega, \varepsilon, T)$ {Equation (3)}
  end if
  SpikingNeuron.__init__(ϑ, ...) {default initialisation}
  $\beta_{\text{param}} \leftarrow \beta$; $\omega_{\text{param}} \leftarrow \omega \Delta t$ {trainable values}
  buffer $\beta \leftarrow (\beta_{\text{param}})^{\Delta t}$; $\rho \leftarrow \cos(\omega_{\text{param}}) + i \sin(\omega_{\text{param}})$; $\gamma \leftarrow \beta \cdot \rho$ {state variables}

4 SS-EEGNET: SPIKING STREAMING EEGNET

Since no spiking, streaming-capable, end-to-end EEG ER models exist, we introduce **SS-EEGNet** (Figure 4): a Spiking Streaming adaptation of EEGNet (Lawhern et al., 2018). EEGNet provides a transparent starting point for our model, being widely used, lightweight, and generalisable, from which the following adaptations build SS-EEGNet (see Section F for additional detail):

1. **Spatial Input Mapping**: now common practice in EEG ER studies, input channels are spatially mapped onto 2D frames for a simple yet effective performance boost (Shen et al., 2020).
2. **Spiking Activations**: default activation functions (ELU units) are replaced with spiking neurons, providing both more efficient and informative processing. In particular, the first convolutional block performs frequency filtering by leveraging Recurrent RF (RRF) neurons – recurrent neurons feature feedback connections to improve temporal memory – rather than learned convolutional filters.
3. **Step-wise Processing**: to enable real-time, streamed operation, the 2D EEG frames are supplied step-by-step, relying on the dynamics of spiking neurons for temporal information rather than convolutions or pooling operations.

The resulting model retains the core functionality of EEGNet of frequency+spatial filtering followed by filter mixing, but now in a streaming and spiking manner. It should be noted that the output is currently *not streamed*; instead, for a given EEG sample, Leaky Integrate (LI) neurons aggregate signals per output class, with classification based on the highest membrane potential averaged over all time steps. This approach matches typical EEG ER pipelines, though, as we demonstrate in Section 6.3.1, SS-EEGNet can still be set up to produce low-latency streamed output.

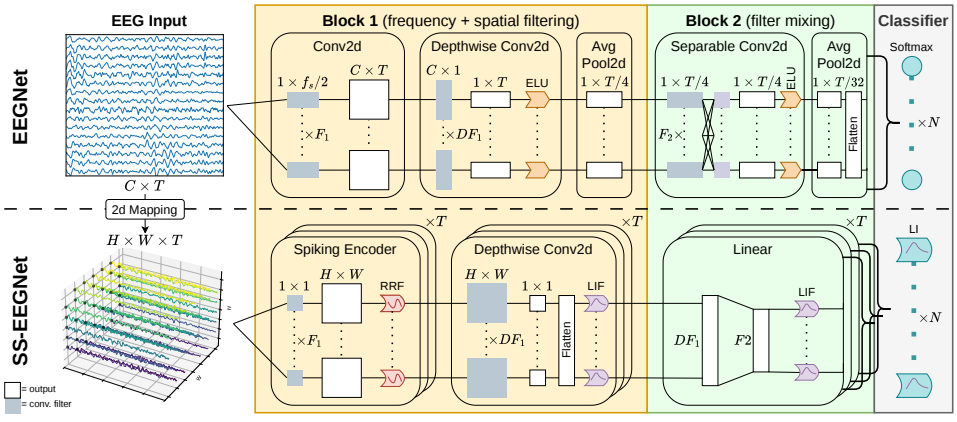


Figure 4: Architecture overview EEGNet (top) vs. SS-EEGNet (bottom). SS-EEGNet matches EEGNet functionality, but now with spatial input mapping, spiking dynamics, and step-wise processing.

5 EXPERIMENTS

5.1 DATASETS

We first compare our RF neuron model against LIF/RLIF, BRF (Higuchi et al., 2024), and DRF (Zhang et al., 2025) – two SOTA RF variants – for performance in three neuromorphic benchmarks. **1) S/PS-MNIST** (Sequential/Permuted Sequential MNIST) consists of MNIST handwritten digit images unrolled into sequences of pixel values, serving as a simple task for comparison between sequential models. PS-MNIST additionally applies a fixed random permutation to the pixel order before streaming making it a harder test of long-range temporal memory. **2) N-MNIST** (Orchard et al., 2015) is an event-based adaptation of MNIST, where a neuromorphic vision sensor captured displayed digits while making saccadic movements. These samples contain limited spectral information (Iyer et al., 2021) and hence may effectively isolate other benefits of RF neurons. **3) SHD** (Cramer et al., 2022) is an audio-to-spike dataset specifically generated for SNNs, in which spoken digits were recorded in English *and* German, and converted to events via an artificial ear model; a core SNN benchmark offering a more challenging testing ground.

For the EEG ER task we consider two datasets which are segmented into non-overlapping 6-/3-seconds windows and mapped onto 2D representations corresponding to electrode positioning (Shen et al., 2020). These datasets are **1) DEAP** (Koelstra et al., 2012), containing 32-channel 128 Hz EEG recordings and **2) SEED-VII** (Jiang et al., 2024) containing 62-channel 200 Hz EEG recordings. Classification is performed over high/low valence and arousal (4-class) for DEAP and discrete emotion categories (7-class) for SEED-VII, based on the membrane potential of the output layer averaged over time. While both datasets contain multi-modal data, we focus on EEG only for consistency. See Section D for examples and further details.

5.2 NETWORK IMPLEMENTATIONS

Simple SNNs were implemented for the neuromorphic benchmarks to isolate RF neuron effects: shallow models for the S-MNIST and SHD tasks matching (Higuchi et al., 2024), and a Convolutional SNN (CSNN) for N-MNIST from (Eshraghian, 2023). Only changes made were to save on training expenses, with our focus on efficiency, as we stuck to the standard `atan` spiking surrogate (rather than the multi/double gaussian) and trained for a smaller number of epochs. Since no spiking end-to-end EEG ER models exist, we introduced **SS-EEGNet** as described in Section 4.

Notably, all (R)RF instances across models employed the derived firing threshold from Equation (3), with the hyperparameters ε and T set according to spectral statistics and/or visual inspection of the input data (full network details in Section E). Each of the neuromorphic benchmarks featured $\varepsilon = 1$, as inputs were (near-)binary, with this value doubled for recurrent neurons, which in absence of sparsity mechanisms leaves room for the recurrent layer to learn post-spike behaviour (as a pseudo-reset). T was set by the timescale of activity bursts in each dataset: short horizontal strokes in (P)S-MNIST ($T = 8$), saccadic movements in N-MNIST ($T = 100$), and ~ 0.1s of speech in SHD ($T = 40$). For SS-EEGNet, ε was set according to the average STFT power per frequency-band, with $T = 0.5$s set empirically, providing enough signal for frequencies starting at 2Hz.

6 RESULTS

6.1 NEUROMORPHIC BENCHMARKS

Table 1 summarises the benchmark results across the SNN benchmarks. Besides test accuracy, we report the number of parameters and operational sparsity, $\mathrm{Sp.} = (1 - \frac{\text{\# measured spikes}}{\text{\# possible spikes}}) \cdot 100$, to gauge theoretical efficiency.

RRF models are competitive with the SOTA, where in the (P)S-MINST and SHD tasks a few accuracy points are traded off for notable efficiency gains. The use of the derived threshold enables sparsity *without the need for explicit sparsity mechanisms*, though this reduction in spikes may also explain some of the accuracy loss. Non-recurrent RF neurons push efficiency further still, matching the RLIF baseline accuracy with a fraction of the parameters, demonstrating how the resonating dynamics can offer cheaper temporal memory than full recurrence. The N-MNISTs results additionally confirm that

Table 1: Neuromorphic benchmark results. Best-in-column per task is **bolded**, $(T = \ldots)$ denotes number of time steps, architecture describes dimensionality of the input and subsequent layers.
* Approximate T as max-length padding used may differ per batch.
** Convolutional SNN: $[2\times34\times34] \rightarrow [\text{Conv}^{12}_{5\times5}\text{-MP}_{2\times2}\text{-Enc}] \rightarrow [\text{Conv}^{32}_{5\times5}\text{-MP}_{2\times2}\text{-LIF}] \rightarrow [\text{FC}_{800\rightarrow10}\text{-LIF}]$.

Task	Model	Architecture	# params.(↓)	Sparsity(↑)	Acc.(↑)
$(T = 25)$	LIF	784-256-10	203,264	82.8% / 82.1%	98.2% / 98.4%
S-MNIST / PS-MNIST $(T = 784)$	RLIF	1-256-10	68,608	91.0% / 90.2%	94.0% / 70.5%
	BRF	1-256-10	68,874	92.0% / 86.1%	99.0% / 95.0%
	DRF	-	155.1K	-	**99.5% / 98.2%**
	RF	1-256-10	**3,328**	**93.8% / 95.9%**	93.5% / 89.1%
	RRF	1-256-10	68,864	93.7% / 94.2%	98.2% / 92.0%
N-MNIST $(T \approx 314)^*$	LIF	*CSNN***	**18,200**	91.2%	96.3%
	RF	*CSNN***	**18,200**	**96.1%**	**97.2%**
SHD $(T = 250)$	RLIF	700-128-20	108,672	86.1%	84.5%
	BRF	700-128-20	108,820	88.7%	92.3%
	DRF	-	155.1K	-	**96.2%**
	RF	700-128-20	**92,416**	**90.3%**	84.9%
	RRF	700-128-20	108,800	**90.3%**	87.2%

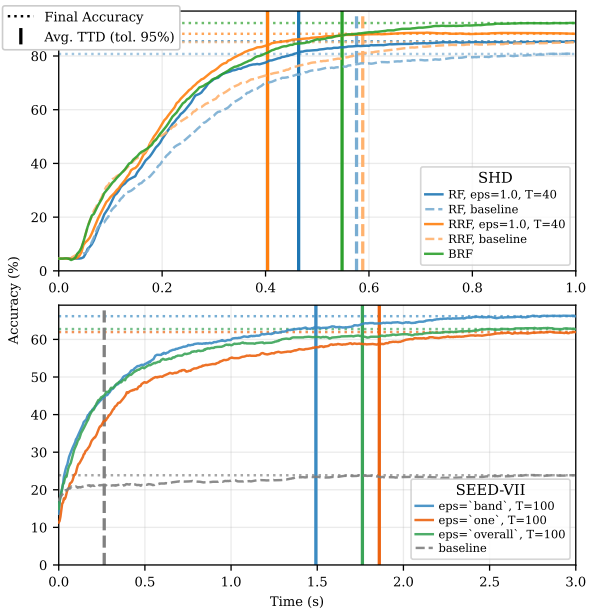
spatial features alone are sufficient to achieve high accuracy in this task, with the RF-based model producing negligible gains ($+0.81\%$) over the LIF baseline. Even so, it clearly shows the inherent sparsity induced by the resonating membrane with the derived threshold: the RF-based model cuts the SOPs/step by more than half despite the light smooth reset employed.

6.2 EEG ER Viability

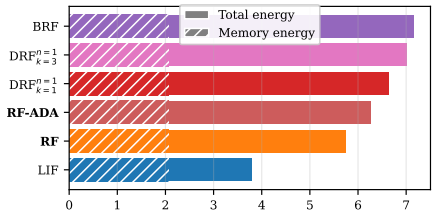
Table 2 presents the SS-EEGNet results on the EEG ER tasks, clearly demonstrating the viability of a spiking and streaming approach, with performance competitive against *end-to-end* SOTA models. Specifically, SS-EEGNet outperforms the EEGNet baseline in DEAP, likely because the latter struggles with the larger window size (6s) – requiring longer temporal memory –and the smaller training set. On SEED-VII, however, SS-EEGNet falls behind in accuracy as EEGNet appears to benefit more from the larger dataset, or rather that the SS-EEGNet configuration left performance on the table. The higher parameter counts of SS-EEGNet models stem from the RF-based encoder, which, while lightweight, requires a larger number of frequency filters (64 vs. 32) to adequately cover the frequency range, driving up the parameter cost of downstream layers – pooling or other dimensionality reductions in intermediate layers may mitigate this overhead in future work. Finally, to clearly isolate the contribution of the RF-based encoder, we ablated neuron type (RLIF vs. RRF) on SEED-VII, where the richer encoder resulted in a substantial accuracy improvement of over 20%.

Table 2: EEG ER results, subject-dependent tests, averaged across subjects.
† Results from literature. * Streaming operation. $^{\rightarrow}$ Sequential operation.

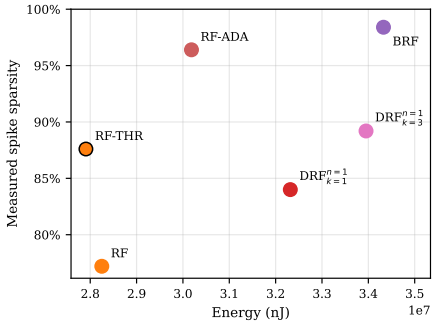
Task	Model	# params.	Acc.			F1		
			Aro.	Val.	Dom.	Aro.	Val.	Dom.
DEAP	HR-SNN†* (Li et al., 2024)	∼5,500	74.8%;	74.6%;	73.5%	-		
	EEGNet-32,2	21,832	73.2%;	68.2%;	71.2%	76.6%;	67.8%;	72.9%
	SS-EEGNet-64,2*$^{\rightarrow}$	28,480	75.5%;	72.0%;	75.6%	78.8%;	75.0%;	79.3%
SEED-VII	MAET† (Jiang et al., 2024)	∼133k	58.1%			55.0%		
	EEGNet-32,2	21,255	79.6%			76.0%		
	SS-EEGNet-64,2$^{*\rightarrow}_{\text{RLIF}}$	32,384	40.1%			34.9%		
	SS-EEGNet-64,2$^{*\rightarrow}_{\text{RRF}}$	32,448	67.5%			67.2%		



(a) Per-step accuracy in SHD (top) and SEED-VII (bottom).



(b) Single step, single neuron energy use (nJ).



(c) Inference energy vs. sparsity %.

Figure 5: Efficiency overview. (a) graphs average SHD and SEED-VII running accuracy for neurons setup with ε, T and without (baselines). The Time-To-Decision (TTD) metric denotes when the model reaches 95% of the final accuracy. (b) showcases single step, single neuron inference energy use simulated within SATA-sim (Yin et al., 2022). Memory energy refers to DRAM/SRAM use, which is static for inference. (c) plots energy use vs. measured sparsity for neurons running a synthetic inference workload in SATA-sim, studying whether induced sparsity offsets computational cost.

6.3 Efficiency Analysis

6.3.1 Faster Response Latency

In support of real-time inference we studied the accuracy over time and measured the Time-To-Decision (TTD) metric in Figure 5a: the earliest time at which the model's predictions comes within $> 95\%$ of the final accuracy. Even without explicitly training for early responses (e.g. via temporal loss), our models reach TTDs of ~ 0.5s for SHD and ~ 1.5s for SEED-VII, halfing the response time, demonstrating how streaming spiking models can be set up to deliver real-time results over approaches reliant on full-window processing.

6.3.2 Lower Energy Use

To estimate neuron energy use, we implemented and ran neuron models through SATA-sim (Yin et al., 2022): a systolic-array based, sparsity-aware training accelerator that models both computation and data movement. For inference, its memory hierarchy (SRAM/DRAM) results in static energy use, with computing energy withing processing elements (the neurons) accounting for most use. Figure 5b shows the resulting single step, single neuron energy usage, where the simpler dynamics of the RF neurons require the least energy across resonating neurons.

6.3.3 Efficient Sparsity Scaling

Building on this, Figure 5c compares mean spike rate against energy use for a synthetic 2-layer convolutional benchmark in SATA-sim, using a resonant input to measure sparsity. RF-based neurons achieve sparsity comparable to other neuron models while remaining efficient. Notably, sparsity has limited impact on energy use in this experiment: the added model complexity rarely pays off, except for RF-THR (RF neuron with derived threshold). These results are implementation/workload-specific; higher accumulate-operation counts would shift the trade-off in favour of increased sparsity.

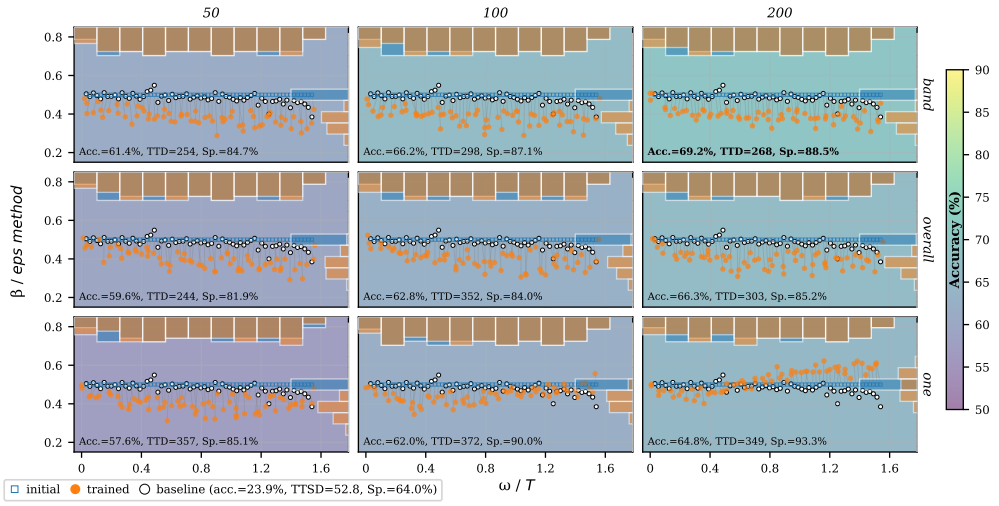


Figure 6: Distributions of β, ω across ε, T configurations, with a baseline in white (no ε, T).

6.4 Robust Parameter Sensitivity

To demonstrate the improved parameter sensitivity of our RF approach, we swept across ε and T configurations for both SHD and SEED-VII tasks, examining performance and the convergence of the trainable β and ω parameters – for sweep details and SHD results, see Section H. Figure 6 confirms the need for properly setting the threshold, as each derived-threshold configuration substantially outperforms the baseline, which falls flat due to excessive spiking. Furthermore, accuracy improves consistently with longer T and more informed ε estimation, suggesting the model effectively translates known spectral structure (e.g. frequency bands) into performance gains.

Trained ω values change little from initialisation, being already set to span the expected frequency range – this stability is not a limitation of the approach, as larger shifts emerge in the SHD sweep – with the exception of some of the lowest values trending towards $\omega = 0$, hinting at some value in LIF-like encoder neurons. Trained β values, by contrast, are consistently lower than initialised, reflecting an overall preference for faster decay, except for the `one` ε method where higher frequencies exhibit larger β to promote spiking; frequencies known to be important for EEG ER.

7 Conclusion

We formulated a robust Resonate-and-Fire neuron, addressing parametrisation challenges, reducing energy use, and enabling reliable use in SNNs through exact integration, principled threshold scaling, and streamlined initialisation. RF neurons achieved competitive accuracy in standard SNN benchmarks while reducing spike operations, parameter counts, and complexity. Building on this, we presented SS-EEGNet, an end-to-end spiking model for real-time EEG-based emotion recognition that pioneers fully streaming operation: recurrent RF neurons are leveraged to extract spectral features directly from raw EEG without windowed (pre)processing. On the DEAP and SEED-VII datasets, SS-EEGNet approaches the performance of end-to-end models, while retaining the low-latency, sparse computation essential for neuromorphic deployment.

Still, several limitations remain. First, all results are simulated, and call for validation on dedicated neuromorphic hardware to confirm the projected efficiency gains. SS-EEGNet itself could also be improved: more expressive surrogate gradients (e.g. multi-Gaussian (Yin et al., 2021)) may aid training but at higher cost, and pooling or broader frequency selectivity in RF encoders could further reduce parameter count. We have also not tested sampling rates beyond 250 Hz, where the precision needed for neuron parameters (ω, β) may exceed what is feasible. Finally, while our TTD results indicate potential responsiveness, genuine streaming deployment remains unvalidated, as the model was trained and evaluated on fixed-length recordings following standard practice. Realising this potential will likely require an online/early-decision training objective alongside a streamed output.

AI USE STATEMENT

In this work, we did not use generative AI tools for any of the required disclosure tasks, except for checking on mistakes/notation in mathematical claims. Additionally, we used generative AI tools for assistance in creating or modifying figures, writing code, summarising or helping understand existing literature, brainstorming, and for editing text to improve readability. All LLM-generated content was manually checked for mistakes and adapted. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Strong efforts have been made to promote reproducibility. Implementation details are included in the appendix, while our code will be made public for publication.

ACKNOWLEDGEMENTS

Thanks to ...

REFERENCES

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, July 2022. ISSN 2162-237X, 2162-2388. doi: 10.1109/TNNLS.2020.3044364.

Maciej Dzieżyc, Martin Gjoreski, Przemysław Kazienko, Stanisław Saganowski, and Matjaž Gams. Can We Ditch Feature Engineering? End-to-End Deep Learning for Affect Recognition from Physiological Sensor Data. *Sensors*, 20(22):6535, November 2020. ISSN 1424-8220. doi: 10.3390/s20226535.

Jason K. Eshraghian. Tutorial 7: Neuromorphic datasets with tonic + snntorch – snntorch documentation. `https://snntorch.readthedocs.io/en/latest/tutorials/tutorial_7.html`, 2023.

Jason K. Eshraghian, Max Ward, Emre Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D. Lu. Training Spiking Neural Networks Using Lessons From Deep Learning, August 2023.

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. ISSN 1534-4320, 1558-0210. doi: 10.1109/TNSRE.2023.3246989.

Zhichao Han, Hongli Chang, Xiaoyan Zhou, Jihao Wang, Lili Wang, and Yongbin Shao. E2ENNet: An end-to-end neural network for emotional brain-computer interface. *Frontiers in Computational Neuroscience*, 16:942979, August 2022. ISSN 1662-5188. doi: 10.3389/fncom.2022.942979.

Saya Higuchi, Sebastian Kairat, Sander M. Bohte, and Sebastian Otte. Balanced Resonate-and-Fire Neurons, October 2024.

Julian Hille, Daniel Auge, Cyprian Grassmann, and Alois Knoll. Resonate-and-Fire Neurons for Radar Interference Detection. In *Proceedings of the International Conference on Neuromorphic Systems 2022*, pp. 1–4, Knoxville TN USA, July 2022. ACM. ISBN 978-1-4503-9789-6. doi: 10.1145/3546790.3546816.

Laxmi R. Iyer, Yansong Chua, and Haizhou Li. Is Neuromorphic MNIST Neuromorphic? Analyzing the Discriminative Power of Neuromorphic Datasets in the Time Domain. *Frontiers in Neuroscience*, 15:608567, March 2021. ISSN 1662-453X. doi: 10.3389/fnins.2021.608567.

Eugene M. Izhikevich. Resonate-and-fire neurons. *Neural Networks*, 14(6-7):883–894, July 2001. ISSN 08936080. doi: 10.1016/S0893-6080(01)00078-8.

Wei-Bang Jiang, Xuan-Hao Liu, Wei-Long Zheng, and Bao-Liang Lu. SEED-VII: A Multimodal Dataset of Six Basic Emotions with Continuous Labels for Emotion Recognition. *IEEE Transactions on Affective Computing*, pp. 1–16, 2024. ISSN 1949-3045. doi: 10.1109/TAFFC.2024.3485057.

S. Koelstra, C. Muhl, M. Soleymani, Jong-Seok Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras. DEAP: A Database for Emotion Analysis ;Using Physiological Signals. *IEEE Transactions on Affective Computing*, 3(1):18–31, January 2012. ISSN 1949-3045. doi: 10.1109/T-AFFC.2011.15.

Dhireesha Kudithipudi, Catherine Schuman, Craig M. Vineyard, Tej Pandit, Cory Merkel, Rajkumar Kubendran, James B. Aimone, Garrick Orchard, Christian Mayr, Ryad Benosman, Joe Hays, Cliff Young, Chiara Bartolozzi, Amitava Majumdar, Suma George Cardwell, Melika Payvand, Sonia Buckley, Shruti Kulkarni, Hector A. Gonzalez, Gert Cauwenberghs, Chetan Singh Thakur, Anand Subramoney, and Steve Furber. Neuromorphic computing at scale. *Nature*, 637(8047):801–812, January 2025. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-024-08253-8.

Vernon J. Lawhern, Amelia J. Solon, Nicholas R. Waytowich, Stephen M. Gordon, Chou P. Hung, and Brent J. Lance. EEGNet: A Compact Convolutional Network for EEG-based Brain-Computer Interfaces. *Journal of Neural Engineering*, 15(5):056013, October 2018. ISSN 1741-2560, 1741-2552. doi: 10.1088/1741-2552/aace8c.

Hendrik M. Lehmann, Julian Hille, Cyprian Grassmann, and Vadim Issakov. Direct Signal Encoding With Analog Resonate-and-Fire Neurons. *IEEE Access*, 11:50052–50063, 2023. ISSN 2169-3536. doi: 10.1109/access.2023.3278098.

Yulin Li, Liangwei Fan, Hui Shen, and Dewen Hu. HR-SNN: An End-to-End Spiking Neural Network for Four-Class Classification Motor Imagery Brain–Computer Interface. *IEEE Transactions on Cognitive and Developmental Systems*, 16(6):1955–1968, December 2024. ISSN 2379-8920, 2379-8939. doi: 10.1109/TCDS.2024.3395443.

Garrick Orchard, Ajinkya Jayawant, Gregory K. Cohen, and Nitish Thakor. Converting Static Image Datasets to Spiking Neuromorphic Datasets Using Saccades. *Frontiers in Neuroscience*, 9, November 2015. ISSN 1662-453X. doi: 10.3389/fnins.2015.00437.

Garrick Orchard, E. Paxon Frady, Daniel Ben Dayan Rubin, Sophia Sanborn, Sumit Bam Shrestha, Friedrich T. Sommer, and Mike Davies. Efficient Neuromorphic Signal Processing with Loihi 2, November 2021.

Stefan Rotter and Markus Diesmann. Exact digital simulation of time-invariant linear systems with applications to neuronal modeling. *Biological Cybernetics*, 81(5-6):381–402, November 1999. ISSN 0340-1200, 1432-0770. doi: 10.1007/s004220050570.

Ahmed Shaaban, Zeineb Chaabouni, Maximilian Strobel, Wolfgang Furtner, Robert Weigel, and Fabian Lurz. Resonate-and-Fire Spiking Neurons for Target Detection and Hand Gesture Recognition: A Hybrid Approach, May 2024.

Fangyao Shen, Guojun Dai, Guang Lin, Jianhai Zhang, Wanzeng Kong, and Hong Zeng. EEG-based emotion recognition using 4D convolutional recurrent neural network. *Cognitive Neurodynamics*, 14(6):815–828, December 2020. ISSN 1871-4080, 1871-4099. doi: 10.1007/s11571-020-09634-1.

Yi Wang, Zhiyi Huang, Brendan McCane, and Phoebe Neo. EmotioNet: A 3-D Convolutional Neural Network for EEG-based Emotion Recognition. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, Rio de Janeiro, July 2018. IEEE. ISBN 978-1-5090-6014-6. doi: 10.1109/IJCNN.2018.8489715.

FeiFan Xu, Deng Pan, Haohao Zheng, Yu Ouyang, Zhe Jia, and Hong Zeng. EESCN: A novel spiking neural network method for EEG-based emotion recognition. *Computer Methods and Programs in Biomedicine*, 243:107927, January 2024. ISSN 01692607. doi: 10.1016/j.cmpb.2023.107927.

Bojian Yin, Federico Corradi, and Sander M. Bohte. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks, March 2021.

Ruokai Yin, Abhishek Moitra, Abhiroop Bhattacharjee, Youngeun Kim, and Priyadarshini Panda. SATA: Sparsity-Aware Training Accelerator for Spiking Neural Networks, December 2022.

Dehao Zhang, Malu Zhang, Shuai Wang, Jingya Wang, Wenjie Wei, Zeyu Ma, Guoqing Wang, Yang Yang, and Haizhou Li. Dendritic Resonate-and-Fire Neuron for Effective and Efficient Long Sequence Modeling, September 2025.

Qinglin Zhao, Lixin Zhang, Haojie Zhang, Hua Jiang, Kunbo Cui, Zhongqing Wu, Jingyu Liu, Mingqi Zhao, Fuze Tian, and Bin Hu. LSNN Model: A Lightweight Spiking Neural Network-Based Depression Classification Model for Wearable EEG Sensors. *IEEE Transactions on Mobile Computing*, pp. 1–15, 2025. ISSN 1536-1233, 1558-0660, 2161-9875. doi: 10.1109/TMC.2025.3586591.

A Solving the RF Neuron Model

The dynamics of the RF neuron (excl. external input) are governed by the ordinary differential equation (ODE)

$$\frac{dU(t)}{dt} = (b + i\omega)U(t)\,, \tag{$\mathrm{RF}_{\mathrm{ODE}}$}$$

where $b \leq 0$ and $\omega \in [0, \pi]$. We consider two methods for solving this equation.

A.1 Forward Euler Method

With Δt as time step size, we start with

$$\frac{dU(t)}{dt} \approx \frac{U(t + \Delta t) - U(t)}{\Delta t}\,.$$

Substituting this into Equation ($\mathrm{RF}_{\mathrm{ODE}}$) gives

$$\frac{dU(t + \Delta t) - U(t)}{\Delta t} = (b + i\omega)U(t)$$

We isolate the membrane at the following step as

$$\begin{aligned} U(t + \Delta t) &= U(t) + \Delta t(b + i\omega)U(t) \\ &= (1 + \Delta t(b + i\omega))\, U(t) \end{aligned} \tag{A1}$$

A.2 Exact Integration Method

Equation ($\mathrm{RF}_{\mathrm{ODE}}$) is linear with constant coefficients. Separating variables gives

$$\frac{1}{U(t)}\frac{dU(t)}{dt} = b + i\omega.$$

Integrating both sides over the interval $[t, t + \Delta t]$ yields

$$\ln U(t + \Delta t) - \ln U(t) = (b + i\omega)\Delta t.$$

Exponentiating, we obtain the exact discrete-time update

$$U(t + \Delta t) = e^{(b+i\omega)\Delta t}U(t). \tag{A2}$$

B Maximum Amplitude of a Driven RF Neuron

Deriving the maximum amplitude requires considering either the real or imaginary part of the membrane potential, as they have equal amplitudes. To simplify the derivation, we first separate Equation ($\mathrm{RF}_{\mathrm{ODE}}$), now including external input $I(t)$, into its real and imaginary components. For brevity we omit t, writing $dU(t)/dt \equiv U'$; setting $U = x + iy$ then gives

$$\begin{aligned} U' &= (b + i\omega)U + I, \quad I \in \mathbb{R} \\ [x + iy]' &= (b + i\omega)(x + iy) + I \\ x' + iy' &= bx + i\omega x + iby + i^2\omega y + I \\ &= bx - \omega y + i(\omega x + by) + I\,. \end{aligned}$$

Now we can separate this into real and imaginary parts,

$$\begin{cases} x' = bx - \omega y + I, & \text{(real)} \\ y' = \omega x + by, & \text{(imaginary).} \end{cases} \tag{B1}$$

We will first focus on the real part x, from which we derive the second-order ODE by differentiating x':

$$\begin{aligned} x'' &= bx' - \omega y' + I' \\ &= bx' - \omega(\omega x + by) + I' \end{aligned}$$

$$x'' = bx' - \omega^2 x - b\omega y + I' \tag{B2}$$

From Equation (B1) we can also obtain

$$y = \frac{bx - x' + I}{\omega}.$$

Substituting into equation B2:

$$\begin{aligned} x'' &= bx' - \omega^2 x - b\omega\left(\frac{bx - x' + I}{\omega}\right) + I' \\ &= bx' - \omega^2 x - b^2 x + bx' - bI + I' \\ &= 2bx' - (\omega^2 + b^2)x - bI + I'. \end{aligned}$$

Thus, the second-order ODE for x is

$$x'' - 2bx' + (\omega^2 + b^2)x = I' - bI\,. \tag{B3}$$

Sinusoidal Input

For a sinusoidal input $I(t) = A_{\text{in}} \sin(\omega_{\text{in}} t) = \Re\big[A_{\text{in}} e^{i\omega_{\text{in}} t}\big]$, the driven equation of motion becomes

$$\begin{aligned} x'' - 2bx' + (\omega^2 + b^2)x &= [A_{\text{in}} e^{i\omega_{\text{in}} t}]' - b \cdot A_{\text{in}} e^{i\omega_{\text{in}} t} \\ &= A_{\text{in}}(-b + i\omega_{\text{in}}) e^{i\omega_{\text{in}} t}. \end{aligned} \tag{B4}$$

We solve this with an *ansatz*. Since the system is driven sinusoidally, after transients decay the response ends up sinusoidal with the same frequency ω_{in}, but with a different amplitude and phase $\tilde{A}$:

$$x = \tilde{A} e^{i\omega_{\text{in}} t}\,; \qquad x' = i\omega_{\text{in}} \tilde{A} e^{i\omega_{\text{in}} t}\,; \qquad x'' = (i\omega_{\text{in}})^2 \tilde{A} e^{i\omega_{\text{in}} t} = -\omega_{\text{in}}^2 \tilde{A} e^{i\omega_{\text{in}} t}\,.$$

Substituting into equation B4 yields

$$\begin{aligned} (-\omega_{\text{in}}^2 - 2bi\omega_{\text{in}} + (\omega^2 + b^2))\tilde{A} e^{i\omega_{\text{in}} t} &= A_{\text{in}}(-b + i\omega_{\text{in}}) e^{i\omega_{\text{in}} t}, \\ (-\omega_{\text{in}}^2 - 2bi\omega_{\text{in}} + \omega^2 + b^2)\,\tilde{A} &= A_{\text{in}}(-b + i\omega_{\text{in}}). \\ \tilde{A} &= \frac{A_{\text{in}}(-b + i\omega_{\text{in}})}{-\omega_{\text{in}}^2 - 2bi\omega_{\text{in}} + \omega^2 + b^2}. \end{aligned}$$

Amplitude Response

Writing $\tilde{A} = Ae^{i\phi}$, the response amplitude is

$$\begin{aligned} A_{\text{res}} = \left|\tilde{A}\right| &= \left|\frac{A_{\text{in}}(-b + i\omega_{\text{in}})}{-\omega_{\text{in}}^2 - 2bi\omega_{\text{in}} + \omega^2 + b^2}\right| \\ &= \frac{A_{\text{in}}\sqrt{b^2 + \omega_{\text{in}}^2}}{\sqrt{(b^2 + \omega^2 - \omega_{\text{in}}^2)^2 + (-2b\omega_{\text{in}})^2}}. \\ A_{\text{res}} &= \frac{A_{\text{in}}\sqrt{b^2 + \omega_{\text{in}}^2}}{\sqrt{(b^2 + \omega^2 - \omega_{\text{in}}^2)^2 + 4b^2\omega_{\text{in}}^2}} \end{aligned} \tag{B5}$$

Finally, at resonance ($\omega_{\text{in}} = \omega$), the maximum amplitude is

$$A_{\max} = \frac{A_{\text{in}}\sqrt{b^2 + \omega^2}}{\sqrt{b^2(b^2 + 4\omega^2)}}\,. \tag{B6}$$

C Full Reset and Refractory Mechanisms

The added complexity of resonating neurons becomes particularly evident in the sparsity mechanisms, as many more options arise. In Table 3 we define each of the sparsity mechanisms we implemented and provide approximate ratings in terms of sparsity (how much is spiking suppressed), stability (whether oscillation is left undisturbed), and simplicity (computational complexity). Do keep in mind that increased sparsity may come at a cost of neuron expressiveness. Figure C1 showcases these reset mechanisms, with Figure C2 doing the same for the refractory mechanisms.

Table 3: Reset and refractory mechanisms with their formulas and operation counts. Mechanisms **in bold** are recommended, $\mathrm{sgn}(\mathrm{x}) = \mathrm{x}/|\mathrm{x}|,\ \text{if } \mathrm{x} \in \mathbb{C},\ k = \lceil \tau_{\mathrm{ref}} \cdot 2\pi/\omega \rceil$.

Name	$R(U,\ S) = \ldots$ [Reset Function]	Sparsity	Stability	Simplicity
No reset [default]	U	★☆☆	★★★	★★★
Simple smooth reset	$U \cdot \beta^{S \cdot s}, \quad s \in \mathbb{Z}_{\geq 2}$	★☆☆	★★★	★★☆
Smooth reset (Higuchi et al., 2024)	$q[t+1] = q[t] \cdot \beta_q + S \cdot \vartheta$ $U \cdot (1 - q[t+1] \cdot \Delta t)$	★½☆	★★★	★☆☆
Partial soft reset ($\Im$)	$U - S \cdot \mathrm{sgn}(U_{\mathrm{spk}}[t]) \cdot i\vartheta \cdot \beta$	★★☆	★☆☆	★½☆
Partial soft reset ($\Re$)	$U - S \cdot \mathrm{sgn}(U_{\mathrm{spk}}[t]) \cdot \vartheta \cdot \beta$	★★☆	★☆☆	★½☆
Soft reset	$U - S \cdot \mathrm{sgn}(U) \cdot \vartheta \cdot \beta$	★★★	★★☆	½☆☆
Izhikevich ($\Im$) (Izhikevich, 2001)	$U - S \cdot (U - i\vartheta)$	★★½	★½☆	★★½
Izhikevich ($\Re$)	$U - S \cdot (U - \vartheta)$	★★½	★½☆	★★½
Partial hard reset ($\Im$)	$U - S \cdot i \cdot U_{\mathrm{spk}}[t]$	★★★	½☆☆	★★★
Partial hard reset ($\Re$)	$U - S \cdot U_{\mathrm{spk}}[t]$	★★★	½☆☆	★★★
Hard reset	$U - S \cdot U$	★★★	★½☆	★★★
	[Refractory Mechanism]			
Absolute	$\vartheta[t] = \vartheta' + \infty \cdot \mathbf{1}_{r[t]>0}$ $r[t+1] = \max(r[t] - 1, 0) + S \cdot k$	★★½	★★½	★½☆
Conditional	$\vartheta[t] = \vartheta' + \infty \cdot \mathbf{1}_{r[t]>0}$ $r[t+1] = \mathbf{1}_{U[t] \geq \vartheta'} \cdot (r[t] + S)$	★★½	★★½	★½☆
Adaptive (Higuchi et al., 2024)	$\vartheta[t] = \vartheta' + r[t]$ $r[t+1] = r[t] \cdot \tau_{\mathrm{ref}} + S \cdot \vartheta'$	★★☆	★★★	★☆☆

D Datasets

Figure D3 showcases representative samples for each of the data types. The algorithms below describe the preprocessing steps applied.

Algorithm 2 S-MNIST Preprocessing – 54k train, 6k val., 10k test

Input: x {standard MNIST data}
$x \leftarrow$ `ToTensor(x)` {convert to tensor and scale values to [0, 1]}
$x \leftarrow$ `x.view(1, -1)` {flatten to 1×784}
Return: x

E Network Details

Tables 4 to 8 denote the hyperparameter setups for each task. Unmentioned hyperparameters are left to default values (e.g. no reset or refractory mechanisms for RF neurons).

Algorithm 3 N-MNIST Preprocessing – 54k train, 6k val., 10k test

Input: x {from N-MNIST webpage or the Tonic package}
$x \leftarrow \texttt{Denoise}(x, 10\,000\ \mu s)$ {drop events that are spatio-temporally not close enough to others}

$x \leftarrow \texttt{ToTimeFrame}(x, f_s = 1\,000\text{ Hz}, \texttt{binary=False})$ {bin data into $\lfloor(\max(t) - \min(t)) \cdot f_s\rfloor + 1$ bins containing graded spikes $\in \mathbb{Z}_+$}
$x \leftarrow \texttt{torch.from_numpy(x)}$ {tensor conversion}
if training **then**
  $x \leftarrow \texttt{RandomRotation}(x, [-10°, 10°])$ {data augmentation}
end if
$x \leftarrow \texttt{pad}(x, \max(x))$ {max-length padding (per batch)}
Return: x

Algorithm 4 SHD Preprocessing – 7341 train, 815 val., 2264 test

Input: x {from the Tonic package}
$x \leftarrow \texttt{ToTimeFrame}(x, f_s = 250\text{ Hz}, \texttt{binary=True}, \text{trim} = [0, 1])$ {trim data to 1s, bin into f_s bins containing binary spikes $\in \{0, 1\}$}
$x \leftarrow \texttt{torch.as_tensor}(x)\texttt{.swapaxes}(0, 1)$ {tensor conversion + swap sensor and time axes}
Return: x

Algorithm 5 General EEG Preprocessing – per subject p

Require: X_p, Y_p, f_s, cs, ls {cs: channel selection, ls: label selection}
Optional: format = None, $T_b = 0, w =$ None, $s =$ None {baseline duration T_b, window size w, step size s (both in s)}
$X_p \leftarrow X_p$[cs] {select EEG channels}
$Y_p \leftarrow Y_p$[ls] {select labels}
if format is not None **then**
  $Y_p \leftarrow \texttt{format_labels}(Y_p,\ \text{format},\ \theta)$ {binarise labels}
end if
$X_p \leftarrow X_p[..., T_\text{b} \cdot f_s :]$ {removal baseline}
if w is not None **then**
  $w_\text{steps} \leftarrow \lfloor w \cdot f_s \rceil;\ s_\text{steps} \leftarrow \lfloor s \cdot f_s \rceil$ {$s \leftarrow w$ if s is None}
  $X'_p, Y'_p \leftarrow \texttt{window_data}(X_p, Y_p)$ {$(\text{trials}, \ldots, \text{T}) \rightarrow (\text{trials} \cdot \lceil \text{T}/\text{s}_\text{steps} \rceil, \ldots, \text{w}_\text{steps})$}
end if

Algorithm 6 DEAP Preprocessing – 320 train, 80 test per subject p

Require: X_p, Y_p {Pre-processed EEG data from DEAP}
$X_p, Y_p \leftarrow \texttt{Algorithm 5}\ (X_p, Y_p, f_s = 128\text{ Hz},$
  $\text{cs} = \text{DEAP.EEG},\ \text{ls} = [\text{arousal}, \text{valence}, \text{dominance}],$
  $\text{format} = (\text{'int'}, 4.), T_\text{baseline} = 3s, w = 6s)$
$X_p \leftarrow \texttt{bandpass_filter}(X_p, \text{4-45 Hz})$
$X_p^{2d} \leftarrow \texttt{channels_to_2d_map}(X_p, \text{DEAP.EEG}, h = 7, w = 7)$

Algorithm 7 SEED-VII Preprocessing – 3825 train, 450 val., 450 test per subject p

Require: X_p, Y_p {Pre-processed EEG data from SEED-VII}
$X_p, Y_p \leftarrow \texttt{Algorithm 5}\ (X_p, Y_p, f_s = 250\text{ Hz},$
  $\text{cs} = \text{SEEDVII.EEG},\ \text{ls} = [\text{happy}, \text{surprise}, \text{disgust}, \text{fear}, \text{sad}, \text{anger}, \text{neutral}],$
  $\text{format} = (\text{'int'}, 0.), w = 3s)$ {categorical label formatting}
$X_p \leftarrow \texttt{highpass_filter}(X_p, \text{from} = 1\text{ Hz})$
$X_p^{2d} \leftarrow \texttt{channels_to_2d_map}(X_p, \text{SEEDVII.EEG}, h = 9, w = 9)$

Table 4: Hyperparameters S-MNIST task.
† Learnable parameter.

Model	LIF	RLIF	RF	RRF
Network	784-256-10	1-256-10	1-256-10	1-256-10
Learning rate	1e-3			
Criterion	NLL - final membrane			
Batch size	256			
Optimizer	Adam			
LR scheduling	LinearLR			
Epochs	35			
Hyperparam.	$\beta = 0.95$ $\beta_{\text{LI}} = 0.95$	$\beta = \mathcal{N}(0.8, 0.1)^{\dagger}$ reset = `none` $\beta_{\text{LI}} = 1.0$	$\beta = \mathcal{N}(0.8, 0.05)^{\dagger}$ $\omega = \mathcal{U}(15, 50)^{\dagger}$ $\varepsilon = 1;\ T = 8$ ref. = `adaptive`($\tau_{\text{ref}} = 0.7$) $\beta_{\text{LI}} = 0.9$	$\beta = \mathcal{N}(0.8, 0.05)^{\dagger}$ $\omega = \mathcal{U}(15, 50)^{\dagger}$ $\varepsilon = 2;\ T = 8$ $\beta_{\text{LI}} = 0.9$

Table 5: Hyperparameters N-MNIST task.

Model	LIF	RF
Network	CSNN	CSNN
Learning rate	2e-2	
Criterion	MSE - spike rate (correct=0.8, incorrect=0.2)	
Batch size	250	
Optimizer	Adam	
LR scheduling	CosineAnnealingLR	
Epochs	5	
Hyperparam.	spike_grad = `atan()` $\beta_{\text{all}} = 0.5$	spike_grad = `atan()` $\beta_{\text{all}} = 0.5;\ \omega = \mathcal{U}(50, 450)$ $\varepsilon = 1;\ T = 100$ reset = `smooth`

Table 6: Hyperparameters SHD task.
† Learnable parameter.

Model	RLIF	RF	RRF
Network	700-128-20	700-128-20	700-128-20
Learning rate	1e-3		
Criterion	NLL - mean membrane		
Batch size	32		
Optimizer	Adam		
LR scheduling	CosineAnnealingWarmRestarts($\mathrm{T}_0 = 5, \mathrm{T}_{\text{mult}} = 3$)		
Epochs	20		
Hyperparam.	$\beta = 0.95$ $\beta_{\text{LI}} = 0.95$	$\beta = \mathcal{N}(0.3, 0.1)^{\dagger}$ $\beta_{\text{LI}} = 0.9$	$\beta = \mathcal{N}(0.3, 0.1)^{\dagger}$ $\omega = \mathcal{U}(15, 50)^{\dagger}$ $\varepsilon = 1;\ T = 8$ $\beta_{\text{LI}} = 0.9$

Table 7: Hyperparameters SS-EEGNet.
† Learnable parameter.

Model	DEAP	SEED-VII (RLIF)	SEED-VII (RRF)
Network	SS-EEGNet		
Learning rate	1e-3		
Criterion	NLL - mean membrane		
Batch size	25		
Optimizer	Adam		
LR scheduling	CosineAnnealingLR ($\mathrm{T}_{\max} = 35, \mathrm{eta}_{\min} =$ 1e-5)	CosineAnnealingWarmRestarts ($\mathrm{T}_0 = 5, \mathrm{T}_{\mathrm{mult}} = 2$)	
Epochs	35		
Hyperparam.	spike_grad = `atan()` $F1 = 64, D = 2$ $\mathrm{p}_{\mathrm{drop}} = 0.05$ $\beta = \mathcal{N}(0.8, 0.1)^\dagger$ $\omega = \mathcal{U}(4, 45)^\dagger$ Hz $\varepsilon =$ band mean, $T = f_s/2$ $\beta_{\mathrm{LI}} = 0.8$	spike_grad = `atan()` $F1 = 64, D = 2$ $\mathrm{p}_{\mathrm{drop}} = 0.1$ $\beta = \mathcal{N}(0.9, 0.1)^\dagger$ reset = `none` $\beta_{\mathrm{LI}} = 0.9$	spike_grad = `atan()` $F1 = 64, D = 2$ $\mathrm{p}_{\mathrm{drop}} = 0.1$ $\beta = \mathcal{N}(0.9, 0.1)^\dagger$ $\omega = \mathcal{U}(1, 49)^\dagger$ Hz $\varepsilon =$ band mean, $T = f_s/2$ $\beta_{\mathrm{LI}} = 0.9$

Table 8: Hyperparameters EEGNet.
† Learnable parameter.

Model	DEAP	SEED-VII
Network	EEGNet	
Learning rate	1e-3	
Criterion	Cross Entropy	
Batch size	25	
Optimizer	Adam	
LR scheduling	CosineAnnealingLR ($\mathrm{T}_{\max} = 35, \mathrm{eta}_{\min} =$ 1e-5)	CosineAnnealingWarmRestarts ($\mathrm{T}_0 = 5, \mathrm{T}_{\mathrm{mult}} = 2$)
Epochs	35	
Hyperparam.	$F1 = 32, D = 2, F2 = F1 \cdot D$ kernel length $= f_s//2$ depthwise kernel length $= f_s//8$	

Table 9: Architecture EEGNet vs. SS-EEGNet. C = number of channels, T = number of time steps, H, W = height and width, F_1 = number of frequency filters, D = depth multiplier, F_2 = number of pointwise filters, N = number of classes. FixedDropout entails a mask that stays fixed throughout all time points per sample.

EEGNET				**SS-EEGNET**				
Layer	**# Filters**	**Size**	**Output**	**Layer**	**# Filters**	**Size**	**Output**	**Function**
Input			(1, C, T)	Input			(1, H, W)	
Block 1								
Conv2d BatchNorm	F_1	(1, $f_s/2$)	(F_1, C, T)	Conv2d RF	F_1	(1, 1)	(F_1, H, W)	Frequency filtering
DepthwiseConv2d - BatchNorm ELU AvgPool2d Dropout	$D \cdot F_1$	(C, 1)	($D \cdot F_1$, 1, T) (1, 4) ($D{\cdot}F_1$, 1, T/4)	DepthwiseConv2d Flatten LayerNorm LIF - FixedDropout	$D \cdot F_1$	(H, W)	($D \cdot F_1$, 1, 1) ($D \cdot F_1$)	Spatial filtering
Block 2								
SeparableConv2d BatchNorm ELU AvgPool2d Dropout Flatten	$F_2{=}D{\cdot}F_1$	(1, 16) (1, 8) 	(F_2, 1, T/4) (F_2, 1, T/32) ($F_2 \cdot (T/32)$)	Linear LayerNorm LIF - FixedDropout -	$F_2{=}D{\cdot}F_1$		(F_2)	Filter mixing
Classifier								
Linear Softmax	$N{\cdot}F_2{\cdot}(T/32)$		(N)	Linear LI	F_2		(N)	Output

F SS-EEGNET

G EFFICIENCY BREAKDOWN

For the spiking neuron models, we base the FLOP computation on the cost of inference for a single time step, given their respective update rule. These update rules have been unified notation-wise, where U is the membrane potential, S the spiking output, b, ω the decay and rotation constants, β the real-valued decay factor ($\beta = e^{b\Delta t}$), γ the complex-valued update factor ($\gamma = e^{(b+i\omega)\Delta t}$), $W \cdot X$ the weighted input, ϑ the threshold, Δt the time step size, and $\Theta(x) := [x \geq 0]$ as the (free) Heaviside function.

For our $\mathrm{RF}_{\mathrm{exact}}$ model, we choose for no reset with the adaptive refractory mechanism for sparsity, being an averagely costed setup that generalises well across tasks. Binary multiplications ($S[t] \cdot x$) are treated as masks/conditional assignments (`MSK`) rather than floating-point multiplications, and are conservatively costed equivalently to an `ADD`. Other variables are model-specific; those matching unified notations are differentiated via a subscript, e.g. β, β_{adp} for ALIF.

Do recall that complex arithmetic has increased costs, i.e. let $\gamma = a + ib,\ U_t = x + iy$:

$$\begin{aligned}
\gamma \cdot U_t &\Rightarrow (a+ib)\cdot(x+iy) = (ax-by) + i(ay+bx) && \Rightarrow \texttt{4MUL+1ADD+1SUB} = 6 \text{ FLOPs} \\
\gamma + U_t &\Rightarrow (a+ib) + (x+iy) = (a+x) + i(b+y) && \Rightarrow \texttt{2ADD} = 2 \text{ FLOPs} \\
\gamma - U_t &\Rightarrow (a+ib) - (x+iy) = (a-x) + i(b-y) && \Rightarrow \texttt{2SUB} = 2 \text{ FLOPs}
\end{aligned}$$

We take well-established equations of the SOS filter and STFT and derive their FLOP counts from the minimal single-input→single-output case. The SOS filter of order n is implemented as a cascade of n second-order sections (biquads), each applying a degree-2 rational transfer function per time step:

$$H(z) = (b_0 + b_1 z^{-1} + b_2 z^{-2})/(1 + a_1 z^{-1} + a_2 z^{-2}).$$

The STFT applies a Fourier transform to successive windows of T samples according to

$$S[f,t] = \sum_{k=0}^{N-1} x[k] \cdot w[k-t] \cdot e^{-i2\pi fk/N},$$

where w is the window function and N the FFT size (frequency resolution). The minimal case entails a single window (or $T = |X|$), which sets $w[k] = 1$; this equals the standard radix-2 FFT, requiring $5N \log_2 N$ real FLOPs per window, and $5N \log_2(N)/T$ per time step.

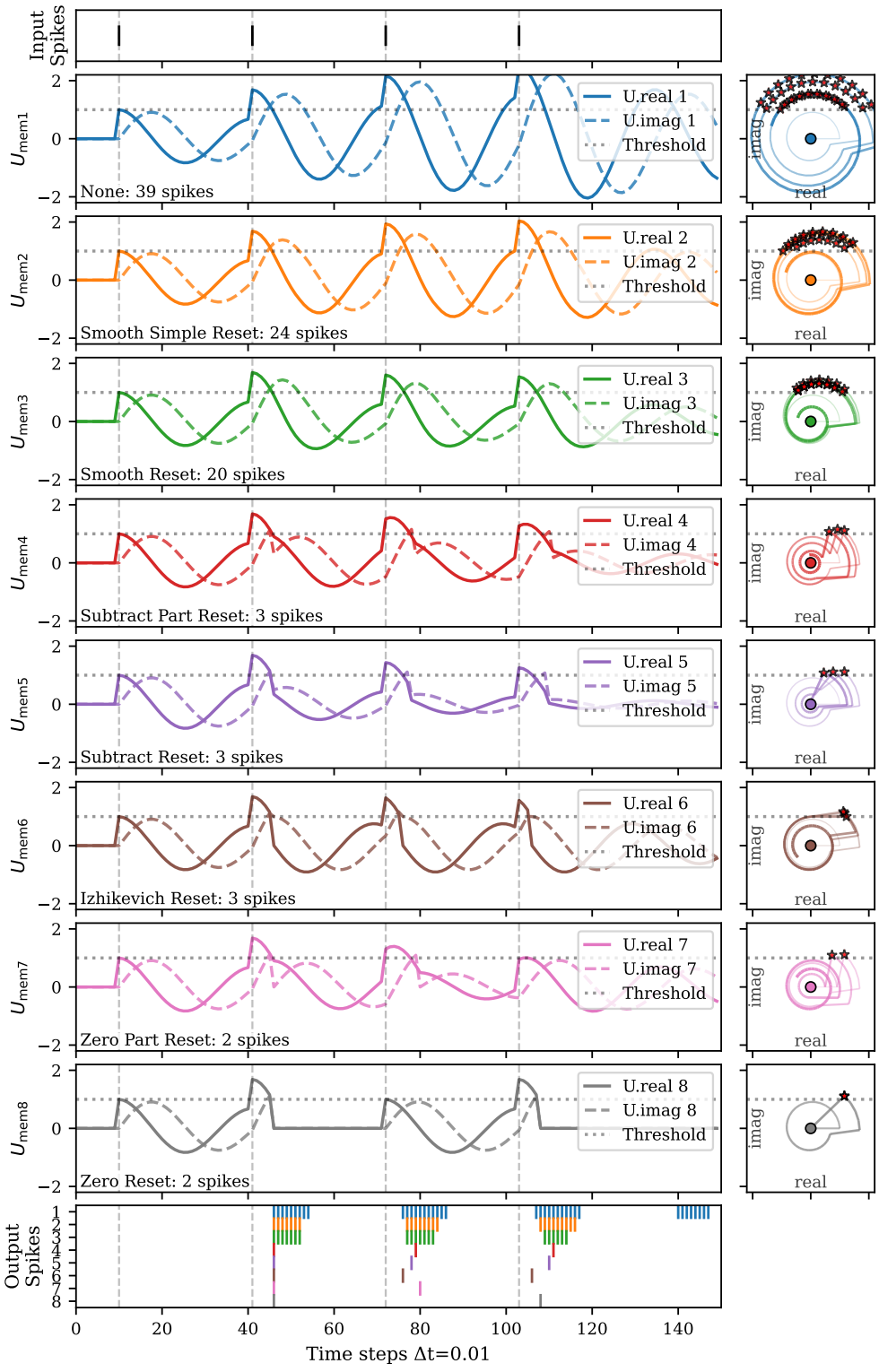


Figure C1: Comparison between reset mechanisms. All other conditions were shared: $\beta = 0.3$, $\omega = 20$, $\Delta t = 0.01$, $U_{\text{spk}} = \Im(U)$, with as input 4 spikes at resonant intervals. The column on the right shows the 'phase portrait' of each U_{mem}, for a geometric display of the neuron's dynamics.

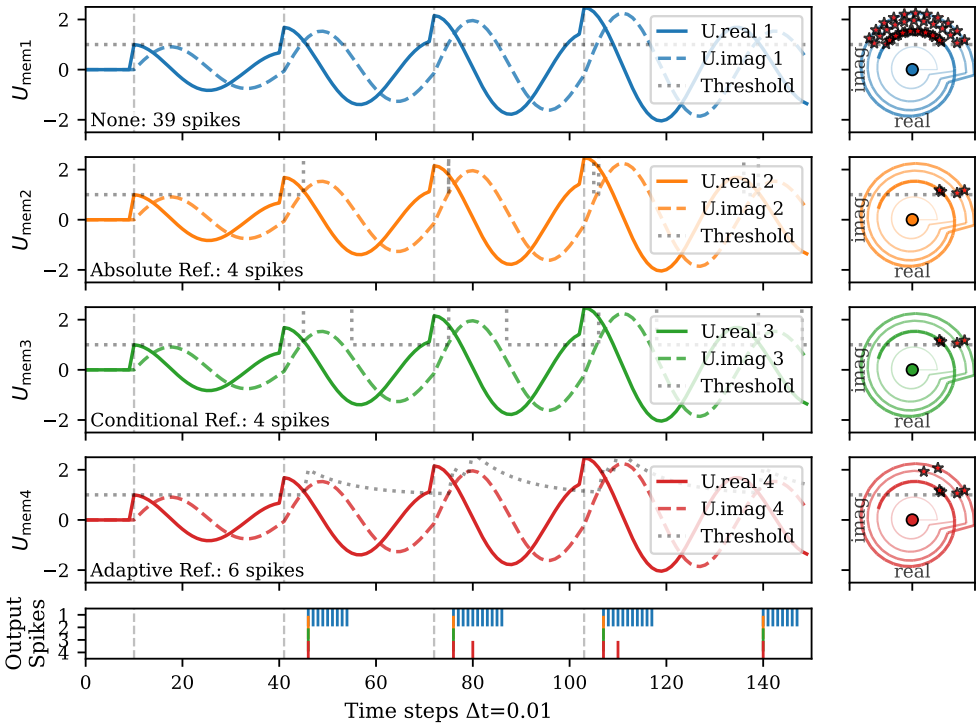
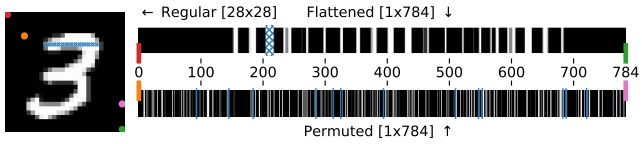
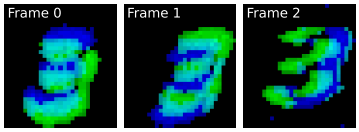


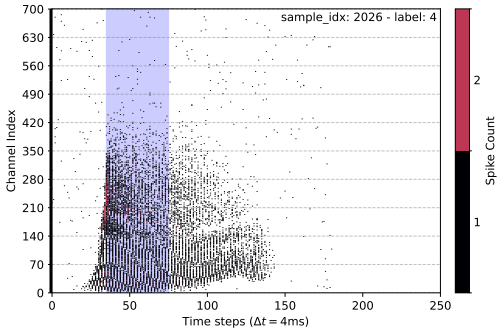
Figure C2: Comparison between refractory mechanisms. Same conditions as in Figure C1.



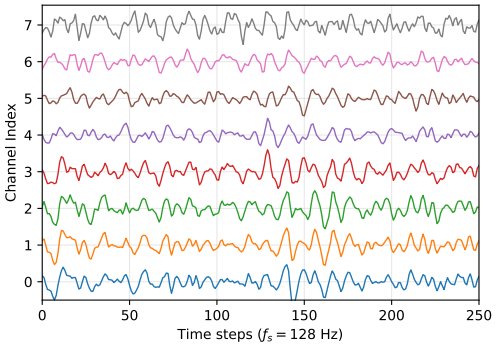
(a) S-MNIST sample. Both regular and flattened forms include one stroke ($T = 13$ steps) highlighted in blue, illustrating the bursty nature of the data.



(b) N-MNIST sample. Frames show accumulated on/off (green/blue) events, each corresponding to saccadic movements of $T = 100$ steps (ms).



(c) SHD sample. Sparse spike trains distributed across channels, forming localized bursts characteristic of speech, here highlighted in blue $T = 40$ steps.



(d) DEAP sample (first 8 channels). Continuous multichannel EEG time series, exhibiting oscillatory temporal dynamics.

Figure D3: Preprocessed samples of each type of dataset, illustrating the differences in structure.

Table 10: The update rule, FLOPs per step, and scaling of neuron models and filters. Variables are real-valued unless notated otherwise. For scaling, C denotes the number of input channels, F the frequency resolution (number of neurons/FFT-size), and T the temporal resolution (number of time steps/windows).
† We provide the real-valued update rule as used in their code over the complex description in the paper (Higuchi et al., 2024) – this does not affect the FLOP total in the end.

Method	Update Rule/Formula	FLOPs/step = Total	Ratio	Scaling
LIF	$S[t] = \Theta(U[t] - \vartheta)$ $\tilde{U}[t+1] = \beta \cdot U[t] + W \cdot X[t+1]$ $U[t+1] = \tilde{U}[t+1] - S[t] \cdot R(\tilde{U}[t+1]), \quad R(U) \in \{U, \vartheta\}$	1SUB 2MUL+1ADD 1MSK+1SUB $= 6$	$1\times$	$\mathcal{O}(CFT)$
ALIF (Yin et al., 2021)	$S[t] = \Theta(U[t] - \vartheta[t])$ $U[t+1] = \beta \cdot U[t] + (1-\beta) \cdot W \cdot X[t+1] - S[t] \cdot \vartheta[t]$ $\eta[t+1] = \rho \cdot \eta[t] + (1-\rho) \cdot S[t], \quad \rho = \exp(-\Delta t/\tau_{\text{adp}})$ $\vartheta[t+1] = \vartheta_0 + \beta_{\text{adp}} \cdot \eta[t+1], \quad \beta = 1.8$ (by default)	1SUB 3MUL+1ADD+2SUB+1MSK 1MUL+1ADD+1SUB+1MSK 1MUL+1ADD $= 14$	$2\frac{1}{3}\times$	$\mathcal{O}(CFT)$
RF$_{\text{euler}}$	$S[t] = \Theta(U_{\text{spk}}[t] - \vartheta), \quad U[t] \in \mathbb{C}$ $\tilde{U}[t+1] = U[t] + \Delta t \cdot ((b + i\omega) \cdot U[t] + W \cdot X[t+1])$ $U[t+1] = \tilde{U}[t+1] - S[t] \cdot R(\tilde{U}[t+1])$	1SUB 7MUL+5ADD 2SUB+2MSK $= 17$	$2\frac{5}{6}\times$	$\mathcal{O}(CFT)$
BRF† (Higuchi et al., 2024)	$b[t+1] = p(\omega) - b' - q[t]$ $U[t+1] = U[t] + \Delta t \cdot (b[t+1] \cdot U[t] - \omega \cdot V[t] + W \cdot X[t+1])$ $V[t+1] = V[t] + \Delta t \cdot (\omega \cdot U[t] + b[t+1] \cdot V[t])$ $S[t+1] = \Theta(\Re(U[t+1]) - \vartheta - q[t])$ $q[t+1] = \gamma_q \cdot q[t] + S[t+1], \quad \gamma_q \in (0,1)$	2SUB 4MUL+2ADD+1SUB 3MUL+2ADD 2SUB 1MUL+1ADD $= 18$	$3\times$	$\mathcal{O}(CFT)$
DRF (Zhang et al., 2025)	$\mathcal{U}[t+1] = \mathcal{D} \cdot \mathcal{U}[t] + W \cdot X[t+1], \quad \mathcal{U}[t] \in \mathbb{C}^n, \quad \mathcal{D} \in \mathbb{C}^{n \times n}$ $H[t+1] = \mathcal{C}^\top \cdot \Re(\mathcal{U}[t+1]), \quad \mathcal{C} \in \mathbb{R}^n,$ for n dendrites $\vartheta[t+1] = \sum_{i=0}^{k} (\alpha_i \cdot S[t-i]) + \vartheta_0, \quad k$ steps history $S[t+1] = \Theta(H[t+1] - \vartheta[t+1])$	(4MUL+2ADD) n+1MUL+1ADD (1MUL+1ADD) n−1ADD (1MSK+1ADD) k 1SUB $= 8n+2k+2$	$\frac{4}{3}n+\frac{2}{3}k+\frac{2}{3}\times$	$\mathcal{O}(C\frac{F}{n}nT)$
RF$_{\text{base}}$	$S[t] = \Theta(U_{\text{spk}}[t] - \vartheta), \quad U[t], \gamma \in \mathbb{C}$ $U[t+1] = \gamma \cdot U[t] + W \cdot X[t+1]$	1SUB 5MUL+3ADD $= 9$	$1\frac{1}{2}\times$	$\mathcal{O}(CFT)$
RF$_{\text{adaptive}}$	$S[t] = \Theta(U_{\text{spk}}[t] - \vartheta - r[t]), \quad U[t], \gamma \in \mathbb{C}$ $r[t+1] = r[t] \cdot \tau_{\text{ref}} + S[t] \cdot \vartheta$ $U[t+1] = \gamma \cdot U[t] + W \cdot X[t+1]$	2SUB 1MUL+1ADD+1MSK 5MUL+3ADD $= 13$	$2\frac{1}{6}\times$	$\mathcal{O}(CFT)$
RF$_{\text{reset}}$	$S[t] = \Theta(U_{\text{spk}}[t] - \vartheta), \quad U[t], \gamma \in \mathbb{C}$ $\tilde{U}[t+1] = \gamma \cdot U[t] + W \cdot X[t+1]$ $U[t+1] = R(\tilde{U}[t+1],\ S[t])$	1SUB 5MUL+3ADD 2SUB+2MSK $= 13$	$2\frac{1}{6}\times$	$\mathcal{O}(CFT)$
SOS Filt.	$y[t] = \sum_{k=0}^{2} b_k \cdot x[t-k] - \sum_{k=1}^{2} a_k \cdot y[t-k],$ for n orders	(5MUL+2ADD+2SUB) $n = 9n$	$1.5n\times$	$\mathcal{O}(CFnT)$
STFT	$S[f] = \sum_{k=0}^{F-1} x[k] \cdot e^{-i2\pi fk/F}$	$5\,F\log_2(F)$	$\frac{5}{6}F\log_2(F)\times$	$\mathcal{O}(CF\log_2(F)T)$

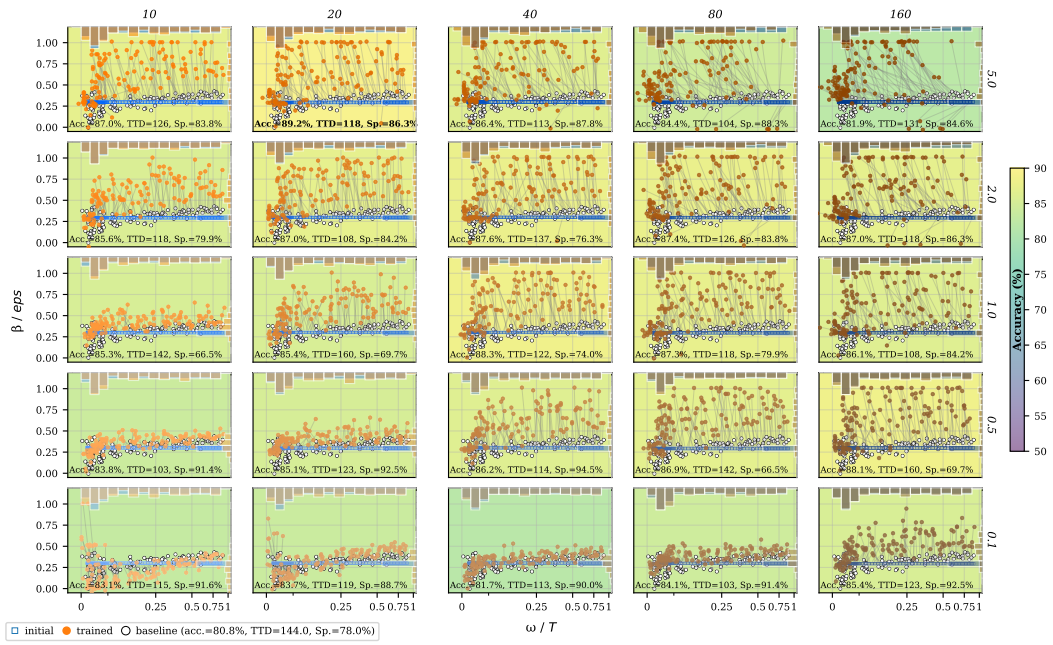


Figure H4: Distributions of β and ω parameters across ε/T configurations, along with a trained baseline in white (no derived threshold).

H Parameter Sensitivity

Figure H4 details the result for the SHD sweep, which featured the same setup as detailed in Table 6 for the RF model, except for:

- $\beta = 0.3$;
- $\varepsilon = [0.1, 0.5, 1.0, 2.0, 5.0]$;
- $T = [10, 20, 40, 80, 160]$;
- no reset or refractory mechanisms.

The SEED sweep followed the same setup detailed in Table 7, except for:

- dataset limited to a subset of subjects 1-5 (due to time constraints);
- 20 epochs with scheduler $T_{\text{mult}} = 3$ (due to time constraints);
- $\beta = 0.5$;
- $\varepsilon = [1, \text{overall mean}, \text{band mean}]$;
- $\text{T} = [1\text{s}, 0.5\text{s}, 0.25\text{s}]$.