A Second-Order SpikingSSM for Wearables

Kartikay Agrawal[1], Abhijeet Vikram[2], Vedant Sharma[2]
Vaishnavi Nagabhushana[1], Ayon Borthakur[1]
[1]Sustain AI Lab, MFSDS&AI, IIT Guwahati, Assam, India
[2]IISER Pune, Maharashtra, India
{a.kartikay,n.vaishnavi,ayon.borthakur}@iitg.ac.in
{vedant.sharma,abhijeet.vikram}@students.iiserpune.ac.in

Abstract

Spiking neural networks have garnered increasing attention due to their energy efficiency, multiplication-free computation, and sparse event-based processing. In parallel, state space models have emerged as scalable alternatives to transformers for long-range sequence modelling by avoiding quadratic dependence on sequence length. We propose SHaRe-SSM (Spiking Harmonic Resonate-and-Fire State Space Model), a second-order spiking SSM for classification and regression on ultra-long sequences. SHaRe-SSM outperforms transformers and first-order SSMs on average while eliminating matrix multiplications, making it highly suitable for resource-constrained applications. To ensure fast computation over tens of thousands of time steps, we leverage a parallel scan formulation of the underlying dynamical system. Furthermore, we introduce a kernel-based spiking regressor, which enables the accurate modelling of dependencies in sequences of up to 50k steps. Our results demonstrate that SHaRe-SSM achieves superior long-range modelling capability with energy efficiency ($52.1\times$ less than ANN-based second-order SSM), positioning it as a strong candidate for resource-constrained devices such as wearables.

1 Introduction

Spike-based deep learning has established itself as an ultra-low-power consumption and sparse computing paradigm for efficient AI in recent years. Spike-based neuromorphic hardware such as Loihi [11, 33], TrueNorth [2], and Dynapse [29] utilises far lower resources than conventional ANN-based designs. Apparently, most spike-based models [40, 41, 22, 32, 35] rely on integrate-and-fire (IF) or leaky IF neurons, which miss key biological traits like oscillations. While the biophysically detailed Hodgkin-Huxley model captures these dynamics, it is computationally prohibitive. Driven by these observations, Resonate-and-fire (RF) neurons [19], computationally as light as IF but more expressive, have gained recent attention [33, 15, 12]. However, RF neurons remain underexplored for very long sequence modelling.

For sequential tasks, transformers are de facto standards [36], but they suffer from quadratic dependence on sequence length. Alternatives, such as KV caching [10] and memory updates [7], reduce overhead but lack the simplicity of RNNs. State space models (SSMs) [14, 34, 13] and their spiking variants [35, 32, 6] bridge this gap. Yet current spiking SSMs struggle with very long sequences.

LinOSS [31, 9], a second-order SSM, addresses this by using stable discretisations with diagonal state matrices, achieving state-of-the-art results on long-range tasks. However, it lacks spike-based communication, which is crucial for energy efficiency [33, 18], such as demanded by a battery-driven wearable. Hence, in this work, we introduce SHaRe-SSM, a second-order spiking SSM designed for

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Learning from Time Series for Health.

extremely long-range tasks and energy-efficient edge AI. Our contributions are: (1) A fully spike-based second-order SSM, without ANN nonlinearities (GeLU, GLU, or GSU). (2) A compatible parallel scan algorithm for fast training and inference. Superior accuracy over first-order SSMs on long-sequence classification, with higher efficiency than ANN-based second-order SSMs. (3) An extension to regression via a convolving kernel, outperforming first-order SSMs on 50K-length tasks.

2 Methods

2.1 Network Description

The Resonate-and-Fire (RF) neuron [19] provides a closer approximation to the Hodgkin-Huxley (HH) model than the Leaky Integrate-and-Fire (LIF) neuron by capturing subthreshold resonance through a 2D linear system with complex eigenvalues. This enables oscillatory dynamics and frequency selectivity, traits absent in LIF but observed in HH neurons. To simplify implementation, second-order real-valued variants such as the Harmonic RF (HRF) [3, 15] reformulate RF as a real harmonic oscillator while preserving oscillatory behaviour without requiring complex initialisation. RF's complex dynamics make it more biophysically realistic, and recent work [15] shows that RF-based models perform best without membrane resets. Building on these insights, we integrate HRF into an SSM framework, compute dynamics linearly via parallel scans, and use a spike function as the activation to propagate spikes. This original HRF formulation is limited to a single neuron. We propose formulating HRF in an SSM framework (SHaRe-SSM). This approach helps in capturing better dynamics. The input spikes are multiplied by a weight matrix, and we propagate spikes from the hidden state $v(t)$. Moreover, removing the damping parameter in such second-order approximations enhances the neuron's ability to capture long-range temporal dependencies by preserving longer oscillations [31]. We define the SHaRe-SSM model by:

$$\begin{aligned} u'(t) &= -\Omega v(t) + Bx(t) \\ v'(t) &= u(t) \\ z(t) &= \Theta(v(t) - \theta_C) \end{aligned} \tag{1}$$

where $u(t), v(t) \in \mathbb{R}^p$ denote the hidden states, $y(t) \in \mathbb{R}^h$ the output, and $x(t) \in \mathbb{R}^h$ the input spike signal. The system is defined by weights $\Omega \in \mathbb{R}^{p \times p}$ which is diagonal, $B \in \mathbb{R}^{p \times h}$, $C \in \mathbb{R}^{h \times p}$, $D \in \mathbb{R}^h$, and an output learnable spiking parameter $\theta_C \in \mathbb{R}^p$ such that spikes are emitted if $v \geq \theta_C$. These thresholds (θ) are learned using a step-double Gaussian surrogate gradient [26, 15]. Since [15] demonstrated that HRF performs well without a reset, we adopted the same strategy here. This eliminates the need for sequential processing, allowing the model to be implemented across time as an activation function. Each SHaRe-SSM block is comprised of a SHaRe-SSM neuron, followed by a linear layer, and a spike function (IF neuron with no reset).

2.2 Encoder and Decoder

For SHaRe-SSM, we design an encoder for encoding the input signal into spike trains (see Algorithm 1). Herein, the inputs every timestep are passed through a learnable linear layer. Finally, an IF neuron with no reset, i.e., a spike function, generates spikes into the SHaRe-SSM block. Notably, this is a data-dependent trainable encoder (without requiring us to specify whether to utilise rate coding or any other encoding variants). Moreover, it utilises neuronal heterogeneity and is instantaneous. The decoder projects the spikes back using a linear Layer with the output dimension equal to the number of classes. For regression, we also propose convolving the output with a learnable filter.

2.3 Discretisation methods

Euler Forward (explicit) discretisation causes divergence over time, making the model unstable. In contrast, IMEX discretisation remains stable and preserves energy, while the IM scheme, though dissipative, is also stable. Below, we analyse the generalizability of IM and IMEX for long sequences. IMEX is particularly promising for regression: as shown in Rusch and Rus [31], removing damping yields a Hamiltonian system [5], where energy conservation is guaranteed under symplectic discretisations. We also evaluate the non-conservative but empirically stable IM scheme. Our formulation adopts a second-order system with position-like state u_n and velocity-like state v_n. The general form

of discrete updates is denoted by:

$$u_n = u_{n-1} + \Delta t(-\Omega v_\star + Bx_n), \tag{2}$$

$$v_n = v_{n-1} + \Delta t\, u_n, \tag{3}$$

$$s_n = Ms_{n-1} + F_n, \tag{4}$$

where the choice of $v_\star$ distinguishes the discretisation schemes: $v_\star = v_n$ for IM (implicit), and $v_\star = v_{n-1}$ for IMEX (implicit-explicit).

Unlike IM, the eigenvalues of M^{IMEX} lie near the unit circle, preserving oscillatory energy over long horizons. This makes IMEX especially suitable for modelling long sequences, where maintaining temporal structure is crucial.

Algorithm 1 SHaRe-SSM Algorithm

Require: Input sequence x
Ensure: N-blocks, spike function Θ , output sequence o
  $x^0 \leftarrow \textit{Encoder}(x)$ {Encode input sequence into spikes}
  for $n = 1, \ldots, N$ **do**
    $z^n \leftarrow$ solution of HRF in (1) with input x^{n-1} via parallel scan aggregated
    $y^n \leftarrow Cz^n + Dx^{n-1}$ {Weighted spike mixing in (1)}
    $y^n \leftarrow \Theta(y^n - \theta_D^n)$
    $y^n \leftarrow Linear(y^n)$
    $y^n \leftarrow \Theta(y^n - \theta^n)$
    $x^n \leftarrow y^n + x^{n-1}$ {Spike mixing}
  end for
  $o \leftarrow \textit{Decoder}(x^N)$ {Decode spikes}

3 Emperical Results

3.1 Human Activity Recognition

Our SNN model is capable of learning multiple states without any sequential recurrence, as it does not have a reset mechanism. Such a design enables parallelisation and reduced energy consumption. Hence, it is well-suited for wearable devices. Although our model is designed for long-range sequence datasets, we also evaluated its performance on Human Activity Recognition(HAR) datasets: UCI-HAR [4] and SHAR datasets [24]. For 30 subjects, UCI-HAR comprises 10.3k instances of 6 activities (walking, walking upstairs/downstairs, sitting, standing, lying) captured with a 3-axis accelerometer and gyroscope (50 Hz) on a Samsung Galaxy SII, while UniMB SHAR includes 11.7k instances of 17 activities (9 daily, 6 fall) recorded with a 3-axis accelerometer (≤ 50 Hz) on a Samsung Galaxy Nexus I9250. SHaRe-SSM achieves an accuracy of 99% and outperforms the best-performing model [23] for UCI-HAR by 0.2%. For SHAR, our model achieves 92.7%, which is better than non-spiking models in terms of energy and performance, and is competitive with spiking models, as it outperforms SpikeDCL by 0.6% and falls behind SpikeDCL by 1.2% (Refer to Table 1).

Model	SNN	UCIHAR	SHAR
CNN	No	96.3±0.1	92.4±0.5
DeepConvLSTM	No	97.9±0.3	90.8±1.0
LSTM	No	82.4±4.0	83.9±0.9
Transformers	No	96.0±0.3	83.2±0.7
SpikeCNN	**Yes**	96.4±0.2	**94.0±0.3**
SpikeDCL	**Yes**	<u>98.9±0.3</u>	92.1±0.8
Ours (IM)	**Yes**	**99.02±0.3**	<u>92.8±0.1</u>
Ours (IMEX)	**Yes**	96.9±0.5	89.1±0.7

Table 1: Performance comparison (Accuracy %) between different networks on Human Activity Recognition datasets: UCI-HAR[4], and SHAR [24]

3.2 Very-Long Range Interactions

Wearable devices operate over long durations, demanding robust prediction on extended sequences. Spiking SSMs, with their high throughput and energy efficiency, are well-suited for this setting. To evaluate performance on very long temporal dependencies, we study the longest benchmark datasets: the Eigenworms dataset (17,984 sequences) for classification and the PPG-DaLiA dataset (49,920 sequences) for regression. Following the hyperparameter protocol of Rusch and Rus [31] and employing Bayesian search [1], we ensure an optimised training. Eigenworms, a subset of the UEA archive, represent C. elegans motion through six eigenworm features to distinguish wild-type behaviour from that of other mutants. PPG-DaLiA involves heart rate prediction from wrist-worn sensor data collected from 15 individuals over 150 minutes at 128 Hz.

For the Long-range classification task on EigenWorms, we observe that our model performs comparably to LinOSS with just 2.2% drop in performance for IM discretisation, and significantly outperforms the IMEX counterpart by 10%. We observe an additional 82.6% energy improvement for the IMEX discretisation (Fig. 1). Our model outperforms all previous models, especially first-order SSMs such as LRU and S6 by 5% and 7.8% respectively.

We present the first regression results of a spiking SSM on extremely long sequences (up to 50k). To address the limited output range of spiking neurons, we introduce a kernel-based spiking regressor with a learnable temporal kernel. As shown in Table 2, SHaRe-SSM models consistently outperform all first-order SSMs. SHaRe-SSM-IMEX surpassing Mamba by 0.016 MSE, demonstrating both the energy-efficient nature of IMEX discretisation and the strong representation power of resonating neurons. Despite their efficiency, our models are only at most 0.027 MSE below second-order SSMs.

We observed from Rusch and Rus [31] that IM discretisation outperforms IMEX for all datasets for classification and vice versa for regression. This trend is observed similarly for our model. Additionally, Hu et al. [17] demonstrated that linear SSM-based models outperform RNN-based models for chaotic systems. Pourcel and Ernoult [28] observed similar trends for the EigenWorms dataset, where the non-linear dynamics from Rusch and Mishra [30] suffer a tragic drop in performance; however, for the PPG dataset, it outperforms its linear counterpart, and only lags behind LinOSS-based models.

Method	Integrator	SNN	EW (ACC ↑)	PPG (MSE ↓)
NRDE[25]		No	83.9 ± 7.3	9.9 ± 1.0
NCDE[20]		No	75.0 ± 3.9	13.5 ± 0.7
Log-NCDE[37]		No	85.6 ± 5.1	9.6 ± 0.6
LRU[27]		No	87.8 ± 2.8	12.2 ± 0.5
S5[34]		No	81.1 ± 3.7	12.6 ± 1.3
S6[13]		No	85.0 ± 16.1	12.9 ± 2.1
Mamba[13]		No	70.9 ± 15.8	10.7 ± 2.2
LinOSS[31]	IM	No	**95.0 ± 4.4**	**7.5 ± 0.5**
Ours	IM	**Yes**	**92.8 ± 3.3**	11.8 ± 0.9
RHEL-Lin[28]	IMEX	No	75.0 ± 9.9	9.5 ± 1.0
RHEL-Nonlin[28]	IMEX	No	50.1 ± 6.7	**8.4 ± 0.5**
LinOSS[31]	IMEX	No	80.0 ± 4.4	**6.4 ± 0.2**
Ours	IMEX	**Yes**	**90.0 ± 5.7**	9.1 ± 0.2

Table 2: Mean and Standard Deviation reported for longest sequences: Accuracy for EigenWorms, and Mean-Squared Error (MSE $\times 10^{-2}$) for PPG-DaLiA dataset across five training runs for the best model (Top three models are highlighted in **red**, **blue**, **violet** respectively).

3.2.1 Energy computation

In this section, we assess and compare the energy computation for LinOSS and SHaRe-SSM for similar hidden-size(H), state-size(P), and num-blocks(N) on the EigenWorms dataset. From Figure 1, we observe that for different state-to-hidden ratios, we get different energy numbers. For the EigenWorms dataset, the average spike rate per time step per block is 0.42. We observe our best performing IMEX model to be $\sim \mathbf{52.1}\times$ energy efficient than an equivalent LinOSS model. Furthermore, since our model also outperforms LinOSS by 10%, it can captures state dynamics better than LinOSS.

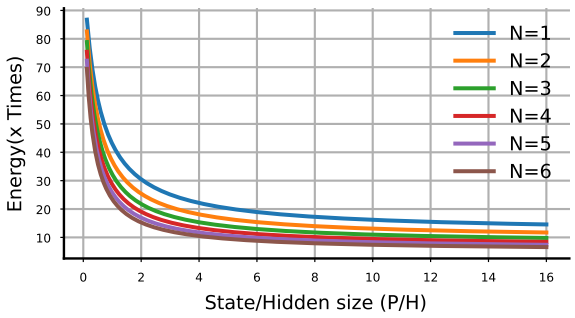

Figure 1: Variation of energy computation of a LinOSS Block [31] to the SHaRe-SSM Block for EigenWorms with respect to State (P), Hidden size (H).

4 Discussion

We propose SHaRe-SSM, an energy-efficient second-order spiking SSM built with harmonic resonate-and-fire neurons, along with a learnable encoder, decoder, and parallel scan method. Unlike prior similar SSMs, SHaRe-SSM is fully spike-based without GeLU/GLU and is tailored for very long-sequence modelling. The model performs well on HAR datasets while consuming significantly less energy, and also achieves superior classification and regression performance on 18k EigenWorms and 50k-length sequences PPG datasets, respectively. Hence, it is ideally suited for use in wearable devices in healthcare. Future work will focus on the deployment of SHaRe-SSM for edge AI using Intel Loihi2 [33].

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631, 2019.

[2] Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, Brian Taba, Michael Beakes, Bernard Brezzo, Jente B. Kuang, Rajit Manohar, William P. Risk, Bryan Jackson, and Dharmendra S. Modha. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(10):1537–1557, 2015. doi: 10.1109/TCAD.2015.2474396.

[3] Badr AlKhamissi, Muhammad ElNokrashy, and David Bernal-Casas. Deep spiking neural networks with resonate-and-fire neurons. *arXiv preprint arXiv:2109.08234*, 2021.

[4] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz, et al. A public domain dataset for human activity recognition using smartphones. In *Esann*, volume 3, pages 3–4, 2013.

[5] V I Arnold. *Mathematical methods of classical mechanics*. Graduate texts in mathematics. Springer, New York, NY, December 2010.

[6] Malyaban Bal and Abhronil Sengupta. P-spikessm: Harnessing probabilistic spiking state space models for long-range dependency tasks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=Sf4ep9Udjf`.

[7] Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *arXiv preprint arXiv:2501.00663*, 2024.

[8] Guy E Blelloch. Prefix sums and their applications. 1990.

[9] Jared Boyer, T. Konstantin Rusch, and Daniela Rus. Learning to dissipate energy in oscillatory state-space models, 2025. URL `https://arxiv.org/abs/2505.12171`.

[10] William Brandon, Mayank Mishra, Aniruddha Nrusimha, Rameswar Panda, and Jonathan Ragan-Kelley. Reducing transformer key-value cache size with cross-layer attention. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=M2UzLRoqic.

[11] Mike Davies, Andreas Wild, Garrick Orchard, Yulia Sandamirskaya, Gabriel A. Fonseca Guerra, Prasad Joshi, Philipp Plank, and Sumedh R. Risbud. Advancing neuromorphic computing with loihi: A survey of results and outlook. *Proceedings of the IEEE*, 109(5):911–934, 2021. doi: 10.1109/JPROC.2021.3067593.

[12] Maxime Fabre, Lyubov Dudchenko, and Emre Neftci. Structured state space model dynamics and parametrization for spiking neural networks. *arXiv preprint arXiv:2506.06374*, 2025.

[13] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[14] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[15] Saya Higuchi, Sebastian Kairat, Sander Bohte, and Sebastian Otte. Balanced resonate-and-fire neurons. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=dkdilv4XD4.

[16] 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)*, pages 10–14. IEEE, 2014.

[17] Zheyuan Hu, Nazanin Ahmadi Daryakenari, Qianli Shen, Kenji Kawaguchi, and George Em Karniadakis. State-space models are accurate and efficient neural operators for dynamical systems. *arXiv preprint arXiv:2409.03231*, 2024.

[18] Nabil Imam and Thomas A Cleland. Rapid online learning and robust recall in a neuromorphic olfactory circuit. *Nat. Mach. Intell.*, 2(3):181–191, March 2020.

[19] Eugene M Izhikevich. Resonate-and-fire neurons. *Neural networks*, 14(6-7):883–894, 2001.

[20] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in neural information processing systems*, 33: 6696–6707, 2020.

[21] Peter M Kogge and Harold S Stone. A parallel algorithm for the efficient solution of a general class of recurrence equations. *IEEE Transactions on Computers*, 22(8):786–793, 1973.

[22] Donghyun Lee, Yuhang Li, Youngeun Kim, Shiting Xiao, and Priyadarshini Panda. Spiking transformer with spatial-temporal attention, 2025. URL https://arxiv.org/abs/2409.19764.

[23] Yuhang Li, Ruokai Yin, Hyoungseob Park, Youngeun Kim, and Priyadarshini Panda. Wearable-based human activity recognition with spatio-temporal spiking neural networks. In *NeurIPS 2022 Workshop on Learning from Time Series for Health*, 2022. URL https://openreview.net/forum?id=MW3Ar-Sun3A.

[24] Daniela Micucci, Marco Mobilio, and Paolo Napoletano. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences*, 7(10):1101, 2017.

[25] James Morrill, Cristopher Salvi, Patrick Kidger, and James Foster. Neural rough differential equations for long time series. In *International Conference on Machine Learning*, pages 7829–7838. PMLR, 2021.

[26] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

[27] Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pages 26670–26698. PMLR, 2023.

[28] Guillaume Pourcel and Maxence Ernoult. Learning long range dependencies through time reversal symmetry breaking. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=w1ihNiIBOc.

[29] Ole Richter, Chenxi Wu, Adrian M Whatley, German Köstinger, Carsten Nielsen, Ning Qiao, and Giacomo Indiveri. Dynap-se2: a scalable multi-core dynamic neuromorphic asynchronous spiking neural network processor. *Neuromorphic computing and engineering*, 4(1):014003, 2024.

[30] T Konstantin Rusch and Siddhartha Mishra. Unicornn: A recurrent model for learning very long time dependencies. In *International Conference on Machine Learning*, pages 9168–9178. PMLR, 2021.

[31] T. Konstantin Rusch and Daniela Rus. Oscillatory state-space models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=GRMfXcAAFh.

[32] Shuaijie Shen, Chao Wang, Renzhuo Huang, Yan Zhong, Qinghai Guo, Zhichao Lu, Jianguo Zhang, and Luziwei Leng. Spikingssms: Learning long sequences with sparse and parallel spiking state space models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39 (19):20380–20388, Apr. 2025. doi: 10.1609/aaai.v39i19.34245. URL https://ojs.aaai.org/index.php/AAAI/article/view/34245.

[33] Sumit Bam Shrestha, Jonathan Timcheck, Paxon Frady, Leobardo Campos-Macias, and Mike Davies. Efficient video and audio processing with loihi 2. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13481–13485, 2024. doi: 10.1109/ICASSP48485.2024.10448003.

[34] Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022.

[35] Matei-Ioan Stan and Oliver Rhodes. Learning long sequences in spiking neural networks. *Scientific Reports*, 14(1):21957, 2024.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[37] Benjamin Walker, Andrew D McLeod, Tiexin Qin, Yichuan Cheng, Haoliang Li, and Terry Lyons. Log neural controlled differential equations: The lie brackets make a difference. *arXiv preprint arXiv:2402.18512*, 2024.

[38] Wenjie Wu, Dexuan Huo, and Hong Chen. Spikf: Spiking fourier network for efficient long-term prediction. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=5jlvLwoO1n.

[39] Weihao Yu, Chenyang Si, Pan Zhou, Mi Luo, Yichen Zhou, Jiashi Feng, Shuicheng Yan, and Xinchao Wang. Metaformer baselines for vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(2):896–912, 2023.

[40] Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng YAN, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=frE4fUwz_h.

[41] Zhaokun Zhou, Jun Niu, Yang Zhang, Li Yuan, and Yuesheng Zhu. Spiking transformer with spatial-temporal spiking self-attention. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10890026.

A Technical Appendix

A.1 Background Theory

To study the theoretical properties of a second-order ODE in SHaRe-SSM, we take inspiration from Rusch and Rus [31]. Our model, like theirs, can be formulated as energy-conserving and possessing dissipative attributes.

A.1.1 Implicit discretisation (IM):

We consider the implicit (backward Euler) discretisation of a second-order system involving a position-like state u_n and a velocity-like state v_n, similar to Rusch and Rus [31]. The implicit scheme is known to introduce additional dissipative terms, which contribute to the stability of the dynamics, particularly in the presence of stiffness.

The discretised updates are given by:

$$\begin{aligned} u_n &= u_{n-1} + \Delta t \left(-\Omega v_n + B x_n\right), \\ v_n &= v_{n-1} + \Delta t \, u_n, \end{aligned}$$

where Ω is a diagonal matrix of oscillation frequencies and B is an input projection matrix. Note that both u_n and v_n are evaluated at the future timestep, in contrast to explicit methods.

Letting the concatenated state be s_n, the above system can be written compactly as:

$$M s_n = s_{n-1} + F_n,$$

where

$$M = \begin{pmatrix} I & \Delta t \Omega \\ -\Delta t I & I \end{pmatrix}, \qquad F_n = \begin{pmatrix} \Delta t B x_n \\ 0 \end{pmatrix}.$$

To obtain an explicit update rule, we algebraically solve the coupled system by introducing the matrix inverse $S = (I + \Delta t^2 \Omega)^{-1}$. Substituting and simplifying yields:

$$s_n = M^{\text{IM}} s_{n-1} + F_n^{\text{IM}}, \tag{5}$$

where

$$M^{\text{IM}} = \begin{pmatrix} S & -S\Delta t \Omega \\ S\Delta t & S \end{pmatrix}, \qquad F_n^{\text{IM}} = \begin{pmatrix} S\Delta t B x_n \\ S\Delta t^2 B x_n \end{pmatrix}.$$

This formulation highlights the stabilising effect of the implicit method: the matrix $S = (I + \Delta t^2 \Omega)^{-1}$ is a Schur complement that acts as a preconditioner, suppressing high-frequency components. Consequently, the eigenvalues of M^{IM} remain bounded within the unit circle for a wide range of Δt, leading to improved numerical stability. The Schur complement can be computed in $\mathcal{O}(m)$ instead of the typical $\mathcal{O}(m^3)$ operations using Gauss-Jordan elimination.

Proposition A.1. *Let $M^{IM} \in \mathbb{R}^{2p \times 2p}$ be the hidden-to-hidden weight matrix of the implicit model SHaRe-SSM-IM* (5)*. We assume that $\Omega_j \geq 0$ for all diagonal elements $j = 1, \ldots, p$ of Ω, and further that $\Delta t > 0$. Then, the complex eigenvalues of M^{IM} are given as,*

$$\lambda_{j_{1,2}} = \frac{1}{1 + \Delta t^2 \Omega_j} \pm \Delta t \frac{\sqrt{\Omega_j}}{1 + \Delta t^2 \Omega_j}$$

with $\lambda_{j_1} = \overline{\lambda_{j_2}}$. Moreover, the spectral radius $\rho(M^{IM})$ is bounded by 1, i.e., $|\lambda_j| \leq 1$ for all $j = 1, \ldots, p$.

Proof. The matrix $M^{\text{IM}} \in \mathbb{R}^{2p\times 2p}$ is defined as

$$M^{\text{IM}} = \begin{bmatrix} S & -\Delta t\, \Omega S \\ \Delta t\, S & S \end{bmatrix},$$

where $S = (I + \Delta t^2 \Omega)^{-1}$, and $\Omega \in \mathbb{R}^{p\times p}$ is diagonal with non-negative entries $\Omega_j \geq 0$ for all $j = 1, \ldots, p$.

To determine the eigenvalues, we compute the characteristic polynomial:

$$\det(M^{\text{IM}} - \lambda I) = \begin{vmatrix} S - \lambda I & -\Delta t\, \Omega S \\ \Delta t\, S & S - \lambda I \end{vmatrix}.$$

Using block Gaussian elimination, we subtract $(\Delta t\, S)(S - \lambda I)^{-1}(-\Delta t\, \Omega S)$ from the lower-right block, giving

$$= \det(S - \lambda I) \cdot \det\left(S - \lambda I + \Delta t^2\, \Omega S^2 (S - \lambda I)^{-1}\right).$$

Since S and Ω are diagonal and commute, this expression decouples elementwise. Let $s_j = \frac{1}{1+\Delta t^2 \Omega_j}$ be the j-th diagonal element of S. Then for each $j = 1, \ldots, p$, the scalar characteristic equation becomes

$$(s_j - \lambda)^2 + \Delta t^2 \Omega_j s_j^2 = 0.$$

Solving this quadratic gives the eigenvalue pair

$$\lambda_{j_{1,2}} = s_j \pm i\, \Delta t\, s_j \sqrt{\Omega_j},$$

where $\lambda_{j_1} = \overline{\lambda_{j_2}}$, i.e., the pair are complex conjugates.

To compute their magnitude:

$$|\lambda_{j_{1,2}}|^2 = s_j^2(1 + \Delta t^2 \Omega_j) = \frac{1}{1 + \Delta t^2 \Omega_j} \leq 1.$$

Hence, all eigenvalues lie on or inside the unit circle in the complex plane, and the spectral radius satisfies $\rho(M^{\text{IM}}) \leq 1$, as claimed. □

Proposition A.2. *Let $\{\lambda_j\}_{j=1}^{2p}$ denote the eigenvalues of the hidden-to-hidden matrix $M^{IM} \in \mathbb{R}^{2p\times 2p}$ of the SHaRe-SSM-IM model* (5)*. Suppose the diagonal entries of $\Omega \in \mathbb{R}^{p\times p}$ are independently drawn as $\Omega_j \sim \mathcal{U}([0, \Omega_{\max}])$, with $\Omega_{\max} > 0$. Then, the N-th moment of the magnitude of the eigenvalues is given by*

$$\mathbb{E}(|\lambda_j|^N) = \frac{(1 + \Delta t^2 \Omega_{\max})^{1-\frac{N}{2}} - 1}{\Delta t^2 \Omega_{\max}(1 - \frac{N}{2})}, \quad \forall j = 1, \ldots, 2p.$$

Proof. From Proposition A.1, each eigenvalue of M^{IM} has magnitude

$$|\lambda_j| = \sqrt{s_j} = \sqrt{\frac{1}{1 + \Delta t^2 \Omega_j}}.$$

The N-th moment of the magnitude is thus.

$$\mathbb{E}(|\lambda_j|^N) = \mathbb{E}\left[\left(\frac{1}{1 + \Delta t^2 \Omega_j}\right)^{\frac{N}{2}}\right].$$

Applying the law of the unconscious statistician and the uniform distribution of $\Omega_j \sim \mathcal{U}([0, \Omega_{\max}])$, we write:

$$\mathbb{E}(|\lambda_j|^N) = \frac{1}{\Omega_{\max}} \int_0^{\Omega_{\max}} (1 + \Delta t^2 x)^{-\frac{N}{2}}\, dx.$$

Substituting $u = 1 + \Delta t^2 x$, so that $du = \Delta t^2\, dx$, the limits change from $x = 0$ to $x = \Omega_{\max}$, corresponding to $u = 1$ to $u = 1 + \Delta t^2 \Omega_{\max}$. The integral becomes:

$$\mathbb{E}(|\lambda_j|^N) = \frac{1}{\Delta t^2 \Omega_{\max}} \int_1^{1+\Delta t^2 \Omega_{\max}} u^{-\frac{N}{2}}\, du.$$

This evaluates to

$$\mathbb{E}(|\lambda_j|^N) = \frac{(1+\Delta t^2\Omega_{\max})^{1-\frac{N}{2}} - 1}{\Delta t^2\Omega_{\max}(1-\frac{N}{2})},$$

□

We can observe from proposition A.2 that even though the spectral radius of eigenvalues is smaller than one (proposition A.1), it is large enough to capture long-range dependencies even for very long-range sequences, even for $\Omega_{\max} = 1, \Delta t = 1$. Hence, we initialize $\Omega_j \sim \mathcal{U}([0,1]), \Delta t_j \sim \mathcal{U}([0,1])$

A.1.2 Implicit-Explicit discretisation (IMEX):

We also utilise an implicit-explicit (IMEX) scheme for discretising the second-order harmonic oscillator system, similar to Rusch and Rus [31]. IMEX methods treat the stiff terms implicitly and the non-stiff or input terms explicitly, resulting in a balanced scheme that enables stable yet undamped oscillations. As shown in Rusch and Rus [31], such schemes preserve the total energy of the system and therefore are particularly well-suited for learning long-range sequential patterns without introducing artificial dissipation.

The update equations under the IMEX discretisation are given by:

$$\begin{aligned} u_n &= u_{n-1} + \Delta t\left(-\Omega v_{n-1} + Bx_n\right), \\ v_n &= v_{n-1} + \Delta t\, u_n, \end{aligned}$$

where the velocity update depends implicitly on the newly computed u_n, while the force term $-\Omega v_{n-1} + Bx_n$ is evaluated using previous state values.

Defining the state vector as s_n, we can rewrite the update in matrix form:

$$Ms_n = M_1 s_{n-1} + F_n,$$

where the matrices M, M_1, and input vector F_n are:

$$M = \begin{pmatrix} I & 0 \\ -\Delta t I & I \end{pmatrix}, M_1 = \begin{pmatrix} I & -\Delta t\Omega \\ 0 & I \end{pmatrix}, F_n = \begin{pmatrix} \Delta t B x_n \\ 0 \end{pmatrix}.$$

Multiplying both sides by M^{-1} yields the closed-form update:

$$s_n = M^{\text{IMEX}} s_{n-1} + F_n^{\text{IMEX}}, \tag{6}$$

where the transition matrix and input vector are given by:

$$M^{\text{IMEX}} = \begin{pmatrix} I & -\Delta t\Omega \\ \Delta t I & I - \Delta t^2\Omega \end{pmatrix}, \qquad F_n^{\text{IMEX}} = \begin{pmatrix} \Delta t B x_n \\ \Delta t^2 B x_n \end{pmatrix}.$$

Proposition A.3. *Let $M^{IMEX} \in \mathbb{R}^{2p\times 2p}$ be the hidden-to-hidden weight matrix of the implicit-explicit model SHaRe-SSM-IMEX* (6)*. Suppose that $\Omega \in \mathbb{R}^{p\times p}$ is diagonal with strictly positive entries $\Omega_j > 0$ for all $j = 1, \ldots, p$, and that the time step satisfies $0 < \Delta t \leq \max_j \left(\frac{2}{\sqrt{\Omega_j}}\right)$. Then, the eigenvalues of M^{IMEX} are given by*

$$\lambda_{j_{1,2}} = \frac{1}{2}(2 - \Delta t^2\Omega_j) \pm \frac{1}{2}\sqrt{\Delta t^2\Omega_j(4 - \Delta t^2\Omega_j)},$$

with $\lambda_{j_1} = \overline{\lambda_{j_2}}$. Moreover, the eigenvalues lie on the complex unit circle, i.e., $|\lambda_j| = 1, \quad \forall j = 1, \ldots, p$.

Proof. The matrix $M^{\text{IMEX}} \in \mathbb{R}^{2p\times 2p}$ has the block form

$$M^{\text{IMEX}} = \begin{bmatrix} I & -\Delta t\,\Omega \\ \Delta t\, I & I \end{bmatrix} \begin{bmatrix} (I + \Delta t^2\Omega)^{-1} & 0 \\ 0 & (I + \Delta t^2\Omega)^{-1} \end{bmatrix},$$

So the effective system matrix becomes

$$M^{\text{IMEX}} = \begin{bmatrix} S & -\Delta t\,\Omega S \\ \Delta t\, S & S \end{bmatrix}, \quad \text{where } S = (I + \Delta t^2\Omega)^{-1}.$$

We analyse the characteristic polynomial:

$$\det(M^{\text{IMEX}} - \lambda I) = \begin{vmatrix} S - \lambda I & -\Delta t\, \Omega S \\ \Delta t\, S & S - \lambda I \end{vmatrix}.$$

Using block elimination, we simplify:

$$= \det(S - \lambda I)^2 + \Delta t^2 \Omega S^2.$$

Since all matrices are diagonal, the problem decouples elementwise. Let $s_j = \frac{1}{1+\Delta t^2 \Omega_j}$ for each $j = 1, \ldots, p$. Then, for each j, the characteristic polynomial becomes

$$\lambda^2 - (2s_j)\lambda + (s_j^2 + \Delta t^2 \Omega_j s_j^2) = \lambda^2 - (2 - \Delta t^2 \Omega_j)\lambda + 1 = 0.$$

Solving this gives the eigenvalue pair

$$\lambda_{j_{1,2}} = \frac{1}{2}(2 - \Delta t^2 \Omega_j) \pm \frac{1}{2}\sqrt{\Delta t^2 \Omega_j (4 - \Delta t^2 \Omega_j)}.$$

To show $|\lambda_{j_{1,2}}| = 1$, we consider two cases:

1. **If** $\Delta t^2 \Omega_j = 4$, then the square root vanishes and
$$\lambda_{j_1} = \lambda_{j_2} = -1, \quad |\lambda_{j_{1,2}}| = 1.$$

2. **If** $\Delta t^2 \Omega_j < 4$, then the eigenvalues are complex conjugates. Their squared magnitude is
$$\begin{aligned} |\lambda_{j_{1,2}}|^2 &= \left(\frac{2 - \Delta t^2 \Omega_j}{2}\right)^2 + \left(\frac{1}{2}\sqrt{\Delta t^2 \Omega_j (4 - \Delta t^2 \Omega_j)}\right)^2 \\ &= \frac{(2 - \Delta t^2 \Omega_j)^2 + \Delta t^2 \Omega_j (4 - \Delta t^2 \Omega_j)}{4} \\ &= \frac{4 - 4\Delta t^2 \Omega_j + \Delta t^4 \Omega_j^2 + 4\Delta t^2 \Omega_j - \Delta t^4 \Omega_j^2}{4} = 1. \end{aligned}$$

Thus, in both cases, $|\lambda_{j_{1,2}}| = 1$, completing the proof. □

A.2 Parallel scan in SHaRe-SSM

Parallel scans [21, 8] exploit associativity to reduce recurrent computation from $\mathcal{O}(N)$ to $\mathcal{O}(\log N)$. Originally developed for RNNs, they have recently been adapted to state-space models [34], enabling efficient architectures such as LRUs [27] and Mamba [13]. In our setting, parallel scans accelerate linear updates, with spike functions applied afterwards. Following Rusch and Rus [31], we define an associative binary operation:

$$(a_1, a_2) \bullet (b_1, b_2) = (b_1 \cdot a_1,\ b_1 \cdot a_2 + b_2), \tag{7}$$

where $\cdot$ denotes matrix-matrix or matrix-vector multiplication. Applying a parallel scan to the input sequence $\{(M, F_n)\}$ efficiently solves

$$s_n = M s_{n-1} + F_n, \tag{8}$$

with the second tuple element storing x_n. Efficiency is achieved by exploiting structured matrices (e.g., diagonal block 2×2 forms in M_{IM} and M_{IMEX}), where each multiplication is linear in the hidden dimension. We use this formulation to implement IM and IMEX discretisations for HRF neurons. Algorithm 1 summarises the SHaRe-SSM implementation.

A.3 Energy Computation

We compute the energy for a LinOSS block and compare it to our SHaRe-SSM block. We can observe that our block doesn't perform any matrix multiplications and is well-suited for neuromorphic hardware. For event-based sensors, we can detach the encoder head and feed data directly to the model for real-time sequential processing. Spike rates from linear layers, SHaRe-SSM neuron and

post weighted spike-mixing layers are given by f^{θ}, f^{θ_C}, f^{θ_D} for Sequence Length(L), State Size(P), Hidden Size(H), respectively.
The ratio of Energy consumed by LinOSS/SHaRe-SSM computed by:

$$\text{Ratio} = \frac{E_{MAC} \times N\left(2LPH + (7+2)LH^2\right)}{E_{AC} \sum_{i=1}^{N} \left(\left(\sum_{j=1}^{i} f_j^{\theta} + f_i^{\theta_C} \right) LPH_{B,C} + f_i^{\theta_D} LH^2 \right)}$$

We estimate the theoretical energy consumption of our model based on prior works [32, 6, 38]. Accordingly, we assume that MAC and AC operations are implemented on a 45nm hardware [16], where $E_{\text{MAC}} = 4.6$pJ and $E_{\text{AC}} = 0.9$pJ. Notably, as discussed in Shen et al., the computational cost of multiplication of a floating-point weight by a binary activation number is represented as an addition-only operation. For ANNs, the theoretical energy consumption of a block n is given by $4.6\text{pJ} \times \text{FLOPs}(n)$. For SNNs, the energy consumption for n is given by $0.9\text{pJ} \times \text{SOPs}(n)$. Calculating theoretical energy consumption requires first calculating the synaptic operations, $\text{SOPs}(n) = f_r \times \text{FLOPs}(n)$, f_r is the firing rate of the input spike train of the block/layer, $\text{FLOPs}(n)$ refers to the number of floating-point operations in layer n, equivalent to the number of multiply-and-accumulate (MAC) operations. SOPs denote the number of spike-based accumulate (AC) operations.

Since Ω is a diagonal matrix, it can be efficiently implemented using parallel scans with computations of order $\mathcal{O}(P \log(L))$ (Total computations are $\mathcal{O}(PL)$), which is negligible. Also, D is $\mathcal{O}(HL)$, which is much smaller than $\mathcal{O}(LPH)$, i.e., computation for B & C matrices or even $\mathcal{O}(LH^2)$, which is for the GLU layer. [31] uses GeLU and GLU non-linearities, which we replace with a linear layer. And, as described in [39], GeLU consumes 14 FLOPs per operation. Moreover, GLU has twice as many FLOPs as a linear layer.

Table 3: Hyperparameters for the Best model for each dataset

Dataset	Method	LR	Hidden	State	Blocks	Time	Kernel
UCI-HAR	IM	1e-3	128	256	2	False	-
UCI-HAR	IMEX	1e-3	128	256	2	False	-
SHAR	IM	1e-3	128	256	2	False	-
SHAR	IMEX	1e-3	128	256	2	False	-
Worms	IM	1e-4	128	256	2	False	-
Worms	IMEX	1e-3	128	64	2	False	-
PPG	IM	1e-3	64	64	6	False	8
PPG	IMEX	1e-3	128	256	6	False	16