SpikeMamba: Spike-Driven State Space Models for Energy-Efficient Biomedical Sequence Modeling

Si Yong Lee[1], Ryangjin Lee[1], Hawon Park[1], Yoora Kim[1], Byungkon Kang[2], and Yoon Seok Yang[2]

[1] Department of Computer Science, Stony Brook University, Stony Brook NY 11794, USA
siyong.lee@stonybrook.edu
[2] Department of Computer Science, The State University of New York Korea, Incheon, Republic of Korea
yoonseok.yang@sunykorea.ac.kr

Abstract. Mamba-based State Space Models (SSMs) provide linear-time sequence modeling but rely on dense floating-point multiply–accumulate (MAC) operations for recurrent updates. In continuous biomedical monitoring, this arithmetic density directly translates into energy cost, limiting practical deployment. SpikeMamba reformulates selective SSM dynamics in the spike domain, rewriting the recurrent update itself as an event-triggered state transition rather than a continuous dense computation. Recurrent dynamics are implemented using Leaky Integrate-and-Fire membrane equations, replacing MAC-heavy transitions with binary accumulations that occur only when spikes are emitted. A lightweight gating mechanism further suppresses updates during temporally inactive segments. Across five biomedical benchmarks, including ECG (PTB-XL) and EEG (CHB-MIT), the model operates below a 15% average firing rate while maintaining accuracy comparable to strong ANN baselines. Energy estimates under a 45 nm CMOS model indicate clear reductions relative to dense ANN-Mamba implementations. Our code is available here.

Keywords: State Space Models · Spiking Neural Networks · Energy Efficiency · Biomedical Signal Processing · Neuromorphic Computing.

1 Introduction

Biomedical time-series analysis, including electrocardiogram (ECG) classification and electroencephalogram (EEG) seizure prediction, requires models that are both temporally expressive and deployable on resource-constrained edge devices [1, 2]. Transformer-based architectures [3, 4] achieve strong empirical performance but scale quadratically with sequence length. State Space Models (SSMs) [5], exemplified by Mamba [6] and Mamba-2 [7], mitigate this limitation

through linear-time selective state dynamics that compress sequential information via discretized recurrent updates, yet remain reliant on dense full-precision arithmetic.

We present SpikeMamba, the first architecture to realize SSMs entirely within the spike domain. SpikeMamba restructures selective recurrence into event-driven state transitions, replacing continuous dense updates with sparse spike-driven dynamics. While Mamba-based models depend on floating-point multiply-and-accumulate (MAC) operations within the selective scan—incurring nontrivial energy overhead for always-on wearable or implantable systems [8]—Spiking Neural Networks (SNNs) [9, 10] offer an alternative paradigm in which information is encoded as sparse binary spikes. This replaces MAC operations with accumulations (ACs) and reduces per-operation energy at comparable CMOS nodes [11]. Neuromorphic processors such as Intel Loihi 2 [12] and spike-driven accelerators like MindCore [13] natively implement Leaky Integrate-and-Fire (LIF) dynamics, enabling direct deployment of spike-based architectures on specialized hardware.

By substituting floating-point MACs with binary accumulations through LIF neurons and a state-dependent gating mechanism, SpikeMamba significantly reduces energy consumption while preserving competitive accuracy across five biomedical benchmarks. Together, these results demonstrate a practical path toward energy-efficient sequential modeling on neuromorphic hardware. The primary contributions are as follows:

1. We express selective SSM dynamics through sparse spike representations that preserve temporal dependencies while eliminating continuous dense updates.
2. We instantiate the SSM recurrence with a LIF neuron, mapping state evolution onto membrane potential dynamics and converting downstream MAC operations into event-triggered accumulations.
3. We introduce a State-Dependent Gate that monitors instantaneous firing activity and membrane potential variance to suppress redundant updates during inactive timesteps.

SpikeMamba is evaluated on five biomedical benchmarks, PTB [17], PTB-XL [18], ADFTD [19], CHB-MIT [20], and MIT-BIH [21], demonstrating competitive predictive performance with substantially reduced energy consumption relative to dense ANN-based implementations.

2 Related Work

State Space Models for Sequence Modeling: Structured State Space Models (S4) [5] provide a linear-complexity alternative to Transformers [22] through discretized recurrent dynamics. Follow-up works include S5 [23] with simplified MIMO state spaces, H3 [24] with multiplicative gating, and Hyena [25] with implicit long convolutions. Mamba [6] introduced input-dependent selective parameters ($\bar{A}$, $\bar{B}$), and Mamba-2 [7] formalized the duality between SSMs and structured attention. Extensions to vision and biomedical imaging include VisionMamba and VMamba [26, 27] as well as UMamba and MambaUNet [28, 29].

Despite improved efficiency, all SSM variants rely on dense floating-point MAC operations at every timestep, limiting deployment on energy-constrained biomedical devices. SpikeMamba reformulates the selective scan into spike-compatible LIF dynamics, preserving $O(N)$ complexity while replacing MACs with event-driven accumulations.

Transformers for Biomedical Time Series: Transformer variants have been adapted to biomedical signals: iTransformer [4], Reformer [30], PatchTST and MTST [31, 32], TimesNet [33], and DLinear [34]. Other efficient designs include Autoformer [35], FEDformer [36], Crossformer [37], and Non-stationary Transformers [38]. Medformer [1] introduces cross-resolution routing and achieves strong results on ECG and EEG benchmarks. Although these models improve predictive accuracy, they uniformly depend on dense MAC arithmetic, with energy cost scaling with attention heads and patch resolutions. SpikeMamba maintains competitive performance while replacing MAC-based computation with AC-based spike dynamics.

Spiking Neural Networks and Energy Efficiency: Spiking Neural Networks (SNNs) encode information as sparse binary spikes, replacing MACs with accumulations (ACs) that reduce per-operation energy at comparable CMOS nodes [11]. Surrogate gradient methods [39,40] enable end-to-end training despite non-differentiable spike functions. Spike-based adaptations of Transformers include Spike-Driven Transformer [14], Spikformer [16], Spike-Driven Transformer V2 [15], and SpikeGPT [41]. While these works demonstrate spike-domain attention models, none reformulate SSM architectures. SpikeMamba addresses this gap by mapping Mamba recurrence onto LIF neuron dynamics and incorporating a state-dependent gate for additional energy savings.

3 Method

3.1 Preliminary: Mamba Selective Scan

Mamba [6] employs a discretized state space model where, at each timestep t, the hidden state $h_t \in \mathbb{R}^D$ is updated as:

$$h_t = \bar{A} \odot h_{t-1} + \bar{B} \odot x_t, \qquad y_t = C \cdot h_t, \tag{1}$$

where $\bar{A}, \bar{B}$ are input-dependent discretized parameters and C is the output projection. This selective scan achieves $O(N)$ complexity but requires dense MAC operations for every timestep.

3.2 SpikeMamba Architecture

SpikeMamba reformulates the Mamba recurrence into the spike domain through three components, illustrated in Fig. 1.

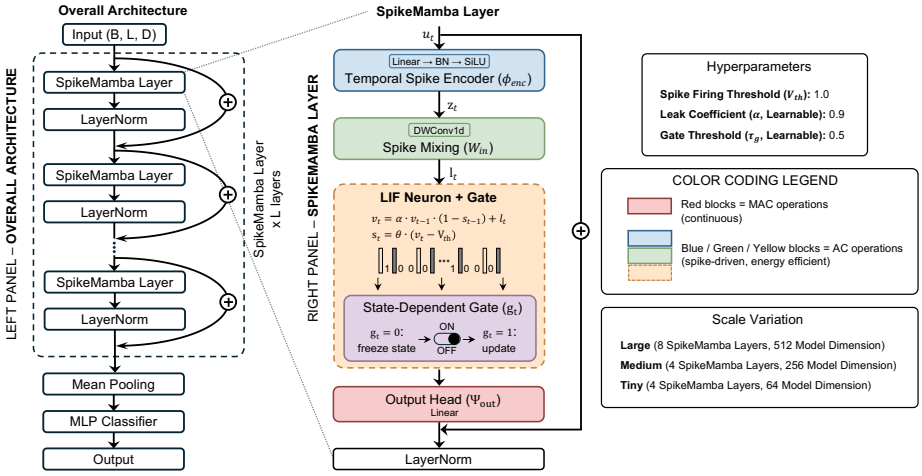

Fig. 1. Overview of the SpikeMamba architecture. Left: The overall model stacks L SpikeMamba layers with residual connections and layer normalization. **Center:** A single SpikeMamba layer: temporal spike encoder ϕ_{enc} (Linear + BN + SiLU), depthwise convolution spike mixing W_{in}, LIF neuron producing binary spikes $s_t \in \{0, 1\}$ (MAC→AC conversion), state-dependent gate g_t, and linear output head ψ_{out}. **Right:** Hyperparameters, MAC/AC color coding, and model scales.

Temporal Spike Encoder ϕ_{enc}. Given a continuous input sequence $u \in \mathbb{R}^{B\times L\times D}$, the encoder projects it into a spike-compatible representation:

$$z_t = \text{SiLU}\big(\text{BN}(W_{\text{enc}} \cdot u_t + b_{\text{enc}})\big), \tag{2}$$

where $W_{\text{enc}} \in \mathbb{R}^{D\times D_{\text{inner}}}$, BN denotes batch normalization [42], and $\text{SiLU}(x) = x\,\sigma(x)$ [43] (σ is the sigmoid) provides the nonlinearity. The BatchNorm layer stabilizes the distribution of pre-spike activations, preventing degenerate firing patterns (all-fire or all-silent).

LIF Neuron with Selective State Update. The core of SpikeMamba replaces the continuous SSM recurrence with a Leaky Integrate-and-Fire (LIF) neuron [44]. After the spike mixing layer W_{in} (implemented as depthwise 1D convolution [45]) generates input current $I_t = W_{\text{in}} * z_t$, the LIF dynamics proceed as:

$$v_t = \alpha \cdot v_{t-1} \cdot (1 - s_{t-1}) + I_t, \tag{3}$$

$$s_t = \Theta(v_t - V_{\text{th}}), \tag{4}$$

$$v_t \leftarrow v_t - s_t \cdot V_{\text{th}}, \tag{5}$$

where $\alpha \in (0, 1)$ is the learnable leak coefficient (parameterized via sigmoid), V_{th} is the firing threshold, $\Theta(\cdot)$ is the Heaviside step function (with surrogate

gradient [39] for backpropagation), and $s_t \in \{0,1\}^D$ is the binary spike output. The term $(1 - s_{t-1})$ implements soft reset: neurons that fired in the previous step have their membrane potential partially cleared.

This LIF formulation is structurally analogous to Eq. 1: the leak factor α plays the role of $\bar{A}$ (state decay), and the input current I_t corresponds to $\bar{B} \odot x_t$ (input injection). The critical difference is that the output s_t is *binary*, transforming all downstream multiplications $w \cdot s_t$ into conditional additions $w \cdot 1 = w$ (when $s_t = 1$) or zero (when $s_t = 0$). Since α is learned per-neuron, it preserves selective SSM state control under binary output sparsification.

State-Dependent Gate: To further reduce computation, SpikeMamba introduces an event-driven gate $g_t \in \{0,1\}$ that determines whether to update the state at timestep t:

$$g_t = \Theta\Big(\sigma\big(W_g \cdot [\bar{s}_t \,\|\, \sigma_v^2(t)] + b_g\big) - \tau\Big), \tag{6}$$

where $\bar{s}_t = \frac{1}{D}\sum_d s_{t,d}$ is the instantaneous spike rate, $\sigma_v^2(t) = \mathrm{Var}(v_t)$ captures membrane potential dynamics, $\sigma(\cdot)$ is the sigmoid function, $W_g \in \mathbb{R}^{2\times 1}$, and τ is a learnable threshold. When $g_t = 0$, the neuron state is *frozen*: $v_t \leftarrow v_{t-1}$, $s_t \leftarrow 0$, eliminating all computation for that timestep.

The complete SpikeMamba layer is then stacked L times with residual connections [46] and layer normalization [47] to form the full model. A final classification head (MLP with mean pooling over time) produces the output logits.

3.3 Energy Model

Following the methodology of Spike-Driven Transformer [14], we analyze energy at 45 nm CMOS technology where $E_{\mathrm{MAC}} = 4.6\,\mathrm{pJ}$ and $E_{\mathrm{AC}} = 0.9\,\mathrm{pJ}$ [11]. For SpikeMamba with $T = 1$ (single forward pass):

$$E_{\mathrm{SpikeMamba}} = \underbrace{E_{\mathrm{MAC}} \cdot \mathrm{FL}_{\mathrm{Input+MLP}}}_{\text{continuous}} + \underbrace{E_{\mathrm{AC}} \cdot R_{\mathrm{eff}} \cdot \mathrm{FL}_{\mathrm{Spike}}}_{\text{spike-driven}}, \tag{7}$$

where $R_{\mathrm{eff}} = R \cdot G$ combines the spike firing rate R and gate activation ratio G.

4 Experiments

4.1 Setup

Datasets: We evaluate on five biomedical time-series benchmarks spanning two modalities. *ECG:* **(1) PTB** [17]: 2-class diagnostic classification (14,552 samples, 12 leads, 500 Hz); **(2) PTB-XL** [18]: 5-class classification (NORM, MI, STTC, CD, HYP; 21,799 samples, 12 leads, 500 Hz); **(3) MIT-BIH** [21]: 5-class arrhythmia classification (N, S, V, F, Q) [48]. *EEG:* **(4) ADFTD** [19]: 3-class Alzheimer's disease classification (A, FTD, CN; 19 channels, 256 Hz);

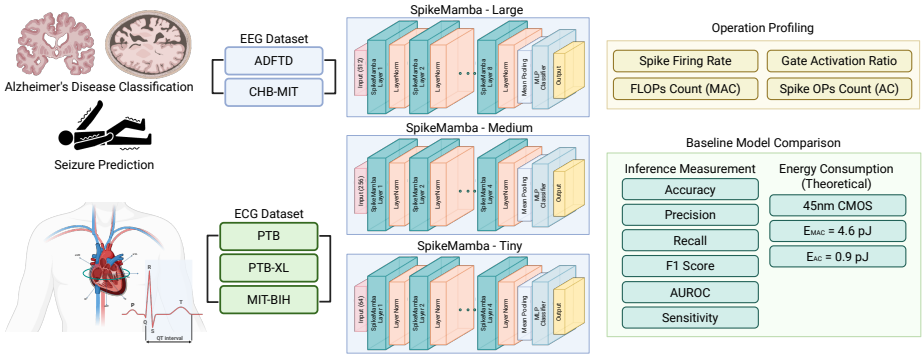

Fig. 2. Experimental Overview. Five biomedical benchmarks are evaluated across three model scales, with operation profiling and energy analysis at 45 nm CMOS technology.

Table 1. Operation breakdown and spike firing rate (SFR) of SpikeMamba variants. $\mathrm{FL}_{\mathrm{Input+MLP}}$ denotes MAC operations from the input projection and classifier; $\mathrm{FL}_{\mathrm{Spike}}$ denotes spike-driven AC operations.

Dataset	Model	SFR	$\mathrm{FL}_{\mathrm{Input+MLP}}$ (M)	$\mathrm{FL}_{\mathrm{Spike}}$ (M)	$\mathrm{FL}_{\mathrm{Total}}$ (M)	Input+MLP (%)	Spike (%)
ADFTD (3-Class)	**SpikeMamba-L**	0.1237	38.8	10821.3	10860.1	0.36	99.64
	SpikeMamba-M	0.1367	14.1	1363.1	1377.3	1.03	98.97
	SpikeMamba-T	0.1271	3.5	89.1	92.7	3.81	96.19
PTB (2-Class)	**SpikeMamba-L**	0.1316	24.7	8454.1	8478.9	0.29	99.71
	SpikeMamba-M	0.1341	8.2	1065.0	1073.2	0.77	99.23
	SpikeMamba-T	0.1409	2.1	69.6	71.7	2.87	97.13
PTB-XL (5-Class)	**SpikeMamba-L**	0.1281	24.7	8454.1	8478.9	0.29	99.71
	SpikeMamba-M	0.1298	8.2	1065.0	1073.2	0.77	99.23
	SpikeMamba-T	0.1589	2.1	69.6	71.7	2.87	97.13

(5) CHB-MIT [20]: seizure prediction from 969 hours of continuous EEG (24 subjects, 198 seizures, 18 bipolar channels at 256 Hz with 5-second segments). Pre-ictal segments are extracted from a 30-minute window ending 5 minutes before seizure onset [49].

Protocol: For PTB, PTB-XL, and ADFTD, we adopt the subject-independent setup from Medformer [1]. For CHB-MIT and MIT-BIH, we use a segment-level random split with a 6:2:2 (train:val:test) ratio.

Model configurations: We evaluate three SpikeMamba scales: *Large* (8 layers, D=512, 8.6M params), *Medium* (4 layers, D=256, 1.1M params), and *Tiny* (4 layers, D=64, 75K params). All variants use the state-dependent gate.

Training: AdamW [50] with learning rate 10^{-3}, weight decay 10^{-4}, cosine annealing [51], and weighted cross-entropy loss. Training runs up to 100 epochs with early stopping (patience = 15). The non-differentiable Heaviside firing function Θ is trained end-to-end via surrogate gradients [39]. All experiments are con-

Table 2. Comparison on PTB, PTB-XL, and ADFTD under the subject-independent setup from Medformer [1]. Best results in **bold**.

Dataset	Model	Params	Acc (%)	Prec (%)	Recall (%)	F1 (%)	AUR (%)	Energy (mJ)
ADFTD (3-Class)	iTransformer	835,331	52.60	46.79	47.28	46.79	67.26	104.32
	MTST	2,731,139	45.60	44.70	45.05	43.31	62.50	22551.78
	Reformer	800,643	50.78	49.64	49.89	47.94	69.17	4974.18
	Medformer	7,975,683	53.27	51.02	**50.71**	**50.65**	**70.93**	11589.69
	SpikeMamba-L	8,634,411	**53.62**	50.04	47.33	46.58	70.04	262.53
	SpikeMamba-M	1,113,111	52.02	**63.55**	44.98	40.01	66.73	41.74
	SpikeMamba-T	75,543	52.28	34.58	43.26	38.16	62.44	**4.08**
PTB (2-Class)	iTransformer	1,081,353	83.89	88.25	76.39	79.06	91.18	51.51
	MTST	2,602,498	76.59	79.88	66.31	67.38	86.86	8803.84
	Reformer	777,602	77.96	81.72	68.20	69.65	91.13	3866.06
	Medformer	5,996,802	83.50	85.19	77.11	79.18	92.81	7564.93
	SpikeMamba-L	8,632,106	**91.72**	**93.23**	94.78	**94.00**	**94.92**	322.91
	SpikeMamba-M	1,111,958	89.00	88.83	**95.98**	92.27	92.73	46.78
	SpikeMamba-T	75,254	80.34	80.61	93.84	86.72	84.35	**4.69**
PTB-XL (5-Class)	iTransformer	834,949	69.28	59.59	54.62	56.20	86.71	51.53
	MTST	2,733,061	72.14	63.84	60.01	61.43	88.97	8807.42
	Reformer	859,653	71.72	63.12	59.20	60.69	88.80	3867.83
	Medformer	7,673,349	72.87	64.14	60.60	62.02	89.66	8225.97
	SpikeMamba-L	8,631,341	**74.16**	**68.42**	59.79	60.91	90.19	260.41
	SpikeMamba-M	1,111,577	74.13	67.10	**60.70**	**62.50**	**90.30**	37.09
	SpikeMamba-T	75,161	72.93	64.92	59.51	61.27	89.26	**3.97**

ducted on a single NVIDIA RTX Pro 6000 96GB Blackwell Workstation GPU using PyTorch 2.6.

Baselines: We compare against iTransformer [4], MTST [32], Reformer [30], and Medformer [1]. Baseline results on PTB, PTB-XL, and ADFTD follow the setup from Medformer [1].

Metrics: Accuracy, Precision, Recall, F1, and AUROC. For CHB-MIT, we additionally report Sensitivity following [52]. All reported results are obtained with 10-run bootstrap resampling. Fig. 2 illustrates the overall experimental framework.

4.2 Energy Efficiency Analysis

Table 1 reports the empirically measured spike firing rates (SFR) and the operation breakdown for all SpikeMamba variants across three datasets. Across all configurations, the SFR remains consistently low, ranging from 0.12 to 0.16, confirming that the LIF neurons in SpikeMamba produce highly sparse binary activations. Notably, spike-driven operations (FL_{Spike}) account for over 96% of the total FLOPs in all cases, reaching up to 99.71% for SpikeMamba-L. Since these spike operations replace energy-expensive multiply-accumulate (MAC) operations with sparse accumulate (AC) operations, only a small fraction of the total computation—the input projection and classifier MLP ($FL_{Input+MLP}$, 0.29–3.81%)—incurs the full MAC energy cost. This extreme sparsity in both firing rate and operation type is the primary driver of SpikeMamba's energy advantage over conventional ANN-based models.

Table 3. Results on CHB-MIT and MIT-BIH (segment-mixed, 6:2:2 split). SpikeMamba results across three model scales.

Dataset	Model	Params	Sens (%)	Acc (%)	Prec (%)	Recall (%)	F1 (%)	AUR (%)	Energy (mJ)
CHB-MIT (2-Class)	Medformer	7,302,786	59.95	83.79	**81.42**	76.52	78.35	88.96	3914.08
	SpikeMamba-L	8,633,642	**89.15**	80.48	60.49	**89.15**	72.01	91.59	320.07
	SpikeMamba-M	1,112,726	81.34	**88.57**	78.92	81.34	**80.02**	**94.48**	56.86
	SpikeMamba-T	75,446	76.73	84.49	70.89	76.73	73.61	91.35	**6.63**
MIT-BIH (5-Class)	Medformer	16,158,213	N/A	**98.99**	**95.43**	95.14	**95.28**	99.76	2147.43
	SpikeMamba-L	8,626,221	N/A	97.05	82.61	94.18	86.37	99.44	366.54
	SpikeMamba-M	1,109,017	N/A	98.87	93.14	**97.01**	94.81	**99.79**	48.19
	SpikeMamba-T	74,521	N/A	86.84	58.73	82.56	64.86	95.94	**4.23**

4.3 Comparison with Baselines

Table 2 presents the comparison of SpikeMamba against four baseline models on PTB, PTB-XL, and ADFTD under the identical subject-independent setting from Medformer [1]. On PTB, SpikeMamba-L achieves the highest accuracy (91.72%) and F1-score (94.00%), outperforming the strongest baseline Medformer (83.50% Acc, 79.18% F1) by a large margin. On PTB-XL, SpikeMamba-M attains the best accuracy (74.13%) and F1 (62.50%), while SpikeMamba-T with only 75K parameters remains competitive with full-scale baselines. On ADFTD, SpikeMamba-L achieves the best accuracy (53.62%) among all models.

Table 3 presents seizure prediction results on CHB-MIT and arrhythmia classification on MIT-BIH. On CHB-MIT, SpikeMamba-L achieves the highest sensitivity (89.15%) and AUROC (91.59%), which are critical for clinical seizure detection, while SpikeMamba-M obtains the best accuracy (88.57%) and AUROC (94.48%). On MIT-BIH, Medformer achieves the best overall performance (98.99% Acc), while SpikeMamba-M remains highly competitive (98.87% Acc, 94.81% F1).

4.4 Theoretical Energy Consumption

The energy columns in Tables 2 and 3 report the theoretical energy consumption at 45 nm technology ($E_{\mathrm{MAC}} = 4.6\,\mathrm{pJ}$, $E_{\mathrm{AC}} = 0.9\,\mathrm{pJ}$), computed via Eq. (7): Vanilla Mamba executes all operations as MACs, whereas SpikeMamba replaces the dominant spike pathway with sparse AC operations scaled by the effective firing rate R_{eff}, while only the input projection and classifier MLP retain the full MAC cost. SpikeMamba-L consumes as low as 260.41 mJ on PTB-XL, over 30× lower than Medformer (8226 mJ) at comparable accuracy while SpikeMamba-M ranges from 37.09 to 56.86 mJ and SpikeMamba-T operates at only 3.97–6.63 mJ across all datasets.

The energy advantage stems from two complementary mechanisms: (1) *spike-driven sparsity* ($R \approx 0.12$–0.16) converts over 96% of MAC operations to sparse AC operations, effectively eliminating ∼84–88% of computations in the spike pathway, and (2) the *state-dependent gate* further modulates computation by selectively skipping inactive timestep updates when the gate produces $g_t \approx 0$.

5 Conclusion

SpikeMamba introduces a novel approach by mapping state space models into the spike domain, fully transforming selective SSM recurrence into event-driven state transitions and replacing continuous dense updates with spike-based dynamics. By employing LIF neurons and a state-dependent gating mechanism instead of floating-point MACs, it substantially reduces energy consumption while preserving strong performance across five biomedical benchmarks. These findings point to a promising direction for energy-efficient sequential modeling and highlight the potential of spike-based architectures for neuromorphic computing. Our energy figures are analytical estimates under a 45 nm CMOS model rather than on-chip measurements. As future work, we plan to deploy SpikeMamba on MindCore [13] to validate real-world energy and latency.

Acknowledgments. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government through the Innovative Human Resource Development for Local Intellectualization program (IITP-2026-RS-2023-00259678), and by RS-2024-0035-8054.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Wang, Y., et al.: Medformer: A multi-granularity patching transformer for medical time-series classification. In: NeurIPS (2024)
2. Craik, A., He, Y., Contreras-Vidal, J.L.: Deep learning for electroencephalogram (EEG) classification tasks: a review. J. Neural Engineering 16(3), 031001 (2019)
3. Zhou, H., et al.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: AAAI, pp. 11106–11115 (2021)
4. Liu, Y., et al.: iTransformer: Inverted transformers are effective for time series forecasting. In: ICLR (2024)
5. Gu, A., Goel, K., Ré, C.: Efficiently modeling long sequences with structured state spaces. In: ICLR (2022)
6. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. In: ICLR (2024)
7. Dao, T., Gu, A.: Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In: ICML (2024)
8. Chen, J., Ran, X.: Deep learning with edge computing: a review. Proceedings of the IEEE 107(8), 1655–1674 (2019)
9. Tavanaei, A., et al.: Deep learning in spiking neural networks. Neural Networks 111, 47–63 (2019)
10. Eshraghian, J.K., et al.: Training spiking neural networks using lessons from deep learning. Proceedings of the IEEE 111(9), 1016–1054 (2023)
11. Horowitz, M.: 1.1 Computing's energy problem (and what we can do about it). In: IEEE ISSCC (2014)
12. Davies, M., et al.: Advancing neuromorphic computing with Loihi: A survey of results and outlook. Proceedings of the IEEE 109(5), 911–934 (2021)

13. Park, H., et al.: MindCore: Spike-driven programmable accelerator for on-device neuromorphic computing. In: IEEE MCSoC (2025)
14. Yao, M., et al.: Spike-driven Transformer. In: NeurIPS (2024)
15. Yao, M., et al.: Spike-driven Transformer V2: Meta spiking neural network. In: ICLR (2024)
16. Zhou, Z., et al.: Spikformer: When spiking neural network meets Transformer. In: ICLR (2023)
17. Goldberger, A.L., et al.: PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource. Circulation 101(23), e215–e220 (2000)
18. Wagner, P., et al.: PTB-XL, a large publicly available electrocardiography dataset. Scientific Data 7(1), 154 (2020)
19. Miltiadous, A., et al.: A dataset of scalp EEG recordings of Alzheimer's disease, frontotemporal dementia and healthy subjects. OpenNeuro (2023)
20. Shoeb, A.H.: Application of machine learning to epileptic seizure onset detection and treatment. PhD thesis, MIT (2009)
21. Moody, G.B., Mark, R.G.: The impact of the MIT-BIH arrhythmia database. IEEE EMBS Magazine 20(3), 45–50 (2001)
22. Vaswani, A., et al.: Attention is all you need. In: NeurIPS (2017)
23. Smith, J.T.H., Warrington, A., Linderman, S.W.: Simplified state space layers for sequence modeling. In: ICLR (2023)
24. Fu, D.Y., et al.: Hungry hungry hippos: Towards language modeling with state space models. In: ICLR (2023)
25. Poli, M., et al.: Hyena hierarchy: Towards larger convolutional language models. In: ICML (2023)
26. Zhu, L., et al.: Vision Mamba: Efficient visual representation learning with bidirectional state space model. In: ICML (2024)
27. Liu, Y., et al.: VMamba: Visual state space model. In: NeurIPS (2024)
28. Ma, J., et al.: U-Mamba: Enhancing long-range dependency for biomedical image segmentation. arXiv:2401.04722 (2024)
29. Wang, Z., et al.: Mamba-UNet: UNet-like pure visual Mamba for medical image segmentation. arXiv:2402.05079 (2024)
30. Kitaev, N., Kaiser, Ł., Levskaya, A.: Reformer: The efficient transformer. In: ICLR (2020)
31. Nie, Y., et al.: A time series is worth 64 words: Long-term forecasting with transformers. In: ICLR (2023)
32. Li, Z., et al.: Multi-scale transformers for time series. arXiv:2310.06625 (2023)
33. Wu, H., et al.: TimesNet: Temporal 2D-variation modeling for general time series analysis. In: ICLR (2023)
34. Zeng, A., et al.: Are transformers effective for time series forecasting? In: AAAI (2023)
35. Wu, H., et al.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In: NeurIPS (2021)
36. Zhou, T., et al.: FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: ICML (2022)
37. Zhang, Y., Yan, J.: Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In: ICLR (2023)
38. Liu, Y., et al.: Non-stationary transformers: Exploring the stationarity in time series forecasting. In: NeurIPS (2022)
39. Neftci, E.O., Mostafa, H., Zenke, F.: Surrogate gradient learning in spiking neural networks. IEEE Signal Processing Magazine 36(6), 51–63 (2019)

40. Rathi, N., et al.: Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation. In: ICLR (2020)
41. Zhu, R., et al.: SpikeGPT: Generative pre-trained language model with spiking neural networks. arXiv:2302.13939 (2023)
42. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML (2015)
43. Elfwing, S., Uchibe, E., Doya, K.: Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural Networks 107, 3–11 (2018)
44. Gerstner, W., Kistler, W.M.: Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press (2002)
45. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: CVPR, pp. 1251–1258 (2017)
46. He, K., et al.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
47. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv:1607.06450 (2016)
48. Kachuee, M., Fazeli, S., Sarrafzadeh, M.: ECG heartbeat classification: A deep transferable representation. In: IEEE ICHI, pp. 443–444 (2018)
49. Daoud, H., Bayoumi, M.A.: Efficient epileptic seizure prediction based on deep learning. IEEE TBME 66(10), 2930–2939 (2019)
50. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
51. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. In: ICLR (2017)
52. Truong, N.D., et al.: Convolutional neural networks for seizure prediction using intracranial and scalp electroencephalogram. Neural Networks 105, 104–111 (2018)