Supplementary Materials: Inherent Redundancy in Spiking Neural Networks

Man Yao[1,2,3*], Jiakui Hu[4,2*], Guangshe Zhao[1†], Yaoyuan Wang[5], Ziyang Zhang[5], Bo Xu[2], Guoqi Li[2†]
[1]School of Automation Science and Engineering, Xi'an Jiaotong University, Xi'an, China
[2]Institute of Automation, Chinese Academy of Sciences, Beijing, China
[3]Peng Cheng Laboratory, Shenzhen, China
[4]Peking University Health Science Center, Peking University, Beijing, China
[5]Advanced Computing and Storage Lab, Huawei Technologies Co Ltd.
manyao@stu.xjtu.edu.cn, jkhu29@stu.pku.edu.cn, zhaogs@mail.xjtu.edu.cn, guoqi.li@ia.ac.cn

S1. Energy Consumption Analysis of attention SNNs

S1.1. Energy Evaluation

In CNN, the times of Floating-point Operations (FLOPs), almost all of which are Multiply-and-Accumulate (MAC), are utilized to determine computational burden. By contrast, the measurement of energy cost for an SNN model is relatively complicated because the FLOPs of the first encoder layer are MAC, while all other Conv or FC layers are synaptic Accumulation (AC). In this work, we employ ASA to optimize the distribution of membrane potential, which reduces spiking firing. Consequently, the energy increase comes from MAC operations due to the regulation of membrane potential. The energy decrease comes from the drop in AC operations caused by sparser spiking firing.

To make a fine-grained evaluation of energy cost, similar to [7, 8], we give some additional definitions as follows: We input all the samples on the test set into the network and count the spike distribution. At timestep t, a Layer's Spiking Firing Rate (L-SFR) is the ratio of spikes produced over all the neurons to the total number of neurons in that layer, and the LASFR is averaging L-SFR across all timesteps T.

The LASFR of the vanilla SNN at n-th Conv and m-th FC layer are Φ_{Conv}^{n} and Φ_{FC}^{m}, respectively. FLOPs of each layer of the SNN are shown in Table S2. Thus the inference energy cost of vanilla SNN E_{Base} is computed as

$$\begin{aligned}E_{Base} &= E_{MAC} \cdot FL_{SNNConv}^{1} + \\ E_{AC} &\cdot \Big(\sum_{n=2}^{N} FL_{SNNConv}^{n} + \sum_{m=1}^{M} FL_{SNNFC}^{m}\Big),\end{aligned} \tag{S1}$$

where N and M are the total number of layers of Conv and FC, E_{MAC} and E_{AC} represent the energy cost of MAC and AC operation, $FL_{SNNConv}^{n}$ and FL_{SNNFC}^{m} are the FLOPs of Conv and FC layer, respectively. Refer to previous SNN works [7, 8, 23, 14, 6], we assume the data for various operations are 32-bit floating point implementation in 45nm technology [4], in which $E_{MAC} = 4.6pJ$ and $E_{AC} = 0.9pJ.$

The additional MAC operations caused by the attention modules can be divided into two parts

$$\Delta_{MAC} = \Delta_{MAC1} + \Delta_{MAC2}, \tag{S2}$$

where Δ_{MAC1} stems from the computation of attention scores, Δ_{MAC2} comes from the regulation operation of the membrane potentials caused by the optimization. Current attention SNNs typically incorporate multiple dimensions of attention modules [22, 11]. In [22], researchers integrate three dimensions of attention, including Temporal Attention (TA), Channel Attention (CA), and Spatial Attention (SA). We compare the ASA module with these modules in terms of Δ_{MAC1} and Δ_{MAC2}, as shown in Table S1.

By optimizing the membrane potential, the attention mechanism drops the spiking activity of SNNs in both Conv and FC layers. We can easily get how much the AC operation in the network has changed by counting the LASFR of the attention SNNs, and the computation formula is shown in column 5 of Table S1.

Then, we can estimate the shift of the energy cost versus the additional computational burden $\Delta_{MAC} = \Delta_{MAC1} + \Delta_{MAC2}$ and the decreased AC operations Δ_{AC} to demonstrate the energy efficiency of the attention SNN. The absolute energy shift between vanilla and attention SNNs can be computed as

$$\Delta_{E} = E_{MAC} \cdot \Delta_{MAC} - E_{AC} \cdot \Delta_{AC}. \tag{S3}$$

We term the attention SNN energy consumption as E_{Att}. Finally, to demonstrate the energy efficiency of the attention

[*] These authors contribute equally to this work
[†] Corresponding author

Attention	Additional Para.(↑)	Additional Computational Complexity		
		Δ_{MAC1} (MAC ↑)	Δ_{MAC2} (MAC ↑)	Δ_{AC} (AC ↓)
TA ([22])	$N \cdot \left(2 \cdot T \cdot \lfloor \frac{T}{r_t} \rfloor\right)$	$N \cdot \left(2 \cdot T \cdot \lfloor \frac{T}{r_t} \rfloor\right)$	$T \cdot N_{Conv-neuron}$	$T \cdot (\sum_{n=1}^{N-1} FL^{n}_{Conv} \cdot \Delta\Phi^{n-1}_{TA-Conv} + \sum_{m=1}^{M} FL^{m}_{FC} \cdot \Delta\Phi^{m-1}_{TA-FC})$
CA ([22])	$\sum_{n=1}^{N} \left(2 \cdot c_n \cdot \lfloor \frac{c_n}{r_c} \rfloor\right)$	$T \cdot \sum_{n=1}^{N} \left(2 \cdot c_n \cdot \lfloor \frac{c_n}{r_c} \rfloor\right)$	$T \cdot N_{Conv-neuron}$	$T \cdot (\sum_{n=1}^{N-1} FL^{n}_{Conv} \cdot \Delta\Phi^{n-1}_{CA-Conv} + \sum_{m=1}^{M} FL^{m}_{FC} \cdot \Delta\Phi^{m-1}_{CA-FC})$
SA ([22])	$N \cdot (2 \cdot 7 \cdot 7)$	$T \cdot \sum_{n=1}^{N} 2 \cdot 7 \cdot 7 \cdot h_n \cdot w_n$	$T \cdot N_{Conv-neuron}$	$T \cdot (\sum_{n=1}^{N-1} FL^{n}_{Conv} \cdot \Delta\Phi^{n-1}_{SA-Conv} + \sum_{m=1}^{M} FL^{m}_{FC} \cdot \Delta\Phi^{m-1}_{SA-FC})$
ASA (**This work**)	$N(\frac{2T^2}{r} + 2 \cdot 2 \cdot 3 \cdot 3)$	$\frac{2NT^2}{r} + \sum_{n=1}^{N} 2 \cdot 2 \cdot 3 \cdot 3 \cdot h_n \cdot w_n$	$T \cdot N_{Conv-neuron}$	$T \cdot (\sum_{n=1}^{N-1} FL^{n}_{Conv} \cdot \Delta\Phi^{n-1}_{ASA-Conv} + \sum_{m=1}^{M} FL^{m}_{FC} \cdot \Delta\Phi^{m-1}_{ASA-FC})$

Table S1: Additional Model and Computational Complexity of various attention modules. Here we assume that each layer of the network uses the attention module. Additional parameters induced by attention modules are very small compared with baseline parameters, which can be ignored. Δ_{MAC1} is caused by the computation of attention weights. Δ_{MAC2} is induced by the refinement of membrane potential, where $N_{Conv-neuron}$ means the number of Conv neurons, T is timestep. Δ_{AC} derives from the drop of network spiking activity, where $\Delta\Phi^{n}_{TA-Conv} = \Phi^{n}_{Conv} - \Phi^{n}_{TA-Conv}$ and $\Delta^{m}_{TA-FC} = \Phi^{m}_{FC} - \Phi^{m}_{TA-FC}$ indicate the shift of LASFR between baseline SNN and TA-SNN in n-th Conv layer and m-th FC layer, respectively. And so on, we can get $\Delta\Phi^{n}_{CA-Conv}$, $\Delta\Phi^{m}_{CA-FC}$, $\Delta\Phi^{n}_{SA-Conv}$, $\Delta\Phi^{m}_{SA-FC}$, $\Delta\Phi^{n}_{ASA-Conv}$ and $\Delta\Phi^{m}_{ASA-FC}$.

Model	FLOPs of a CONV or FC layer		
	Variable	Value	FLOP Type
CNN [12]	FL^{n}_{Conv}	$(k_n)^2 \cdot h_n \cdot w_n \cdot c_{n-1} \cdot c_n$	MAC
	FL^{m}_{FC}	$i_m \cdot o_m$	MAC
SNN [8]	$FL^{n}_{SNNConv}$	$T \cdot FL^{n}_{Conv} \cdot \Phi^{n-1}_{Conv}$	MAC ($n = 1$) or AC ($n > 1$)
	FL^{m}_{SNNFC}	$T \cdot FL^{m}_{FC} \cdot \Phi^{m-1}_{FC}$	AC

Table S2: FLOPs for CNN and SNN models. i_m and o_m are the input and output dimensions of the FC layer, respectively. When the inputs are static images, $\Phi^{0}_{Conv} = 1$. When the inputs are event frames, Φ^{0}_{Conv} is the ratio of non-zero pixels. Moreover, $\Phi^{0}_{FC} = \Phi^{N}_{Conv}$.

Model	Energy (Gesture)	Energy (Gait-day)
Vanilla SNN [21]	1.314mJ	1.502mJ
TCSA-SNN [22]	0.536mJ (-59.2%)	0.582mJ (-61.2%)
ASA-SNN (**Ours**)	0.314mJ (-76.1%)	0.371mJ (-75.3%)

Table S3: The average energy consumption of diverse models on each sample of Gesture and Gait-day. The proportions in parentheses represent r_{REC}.

SNNs, we define the relative energy change ratio r_{REC} as

$$r_{REC} = \frac{\Delta_E}{E_{base}} = \frac{E_{MAC} \cdot \Delta_{MAC} - E_{AC} \cdot \Delta_{AC}}{E_{base}}. \quad \text{(S4)}$$

S1.2. Comparison of Energy Consumption between Attention SNNs

It can be seen from Eq. S3 that only when $\Delta_E < 0$, plugging an additional attention module can reduce energy cost. Thus, we need to try our best to make the benefit (energy reduction by reducing spikes, $E_{AC} \cdot \Delta_{AC}$) outweigh the cost (additional energy consumption from attention, $E_{MAC} \cdot \Delta_{MAC}$). As shown in Table S1, the membrane potential of each neuron at all time steps should be regulated once for each additional attention dimension, i.e., $T \cdot N_{Conv-neuron}$ operations of MAC should be added. In Table S3, we give the average energy consumption of a sample of various models. As can be observed, ASA-SNN consumes the least amount of energy because we only employ the spatial attention module, which lowers Δ_{MAC}.

S2. Datasets and Experimental Setup

S2.1. Datasets

DVS128 Gesture[1]. The DVS128 Gesture dataset is recorded by a DVS128 camera, which has the temporal resolution in µs level and 128×128 spatial resolution and contains 11 kinds of hand gestures from 29 subjects under 3 kinds of illumination conditions. It records 1342 samples of 11 gestures, and each gesture has an average duration of 6 seconds.

DVS128 Gait-day[17]. The DVS128 Gait-day dataset is recorded by a DVS128 camera, which has the temporal resolution in µs level and 128×128 spatial resolution and contains various gaits from 21 volunteers (15 males and 6 females) under 2 kinds of viewing angles. It records 4200 samples, and each gait has an average duration of 4.4 seconds.

DVS128 Gait-night[18]. The DVS128 Gait-night dataset is recorded to investigate if the event camera is able to capture human gaits in low-light conditions. In contrast to Gait-day, Gait-night contains various gaits from 20 volunteers, and each volunteer contributed 200 samples of gait.

DailyAction-DVS[10]. The DailyAction-DVS dataset comprises 1440 samples of 15 subjects acting 12 different actions. Two different lighting setups, including LED light and natural light, were used to record the motions. Under

Model	Architecture-Details
LIF-SNN-3 [21]	Input-MP4
	-64C3
	-128C3-BN-AP2
	-128C3-BN-AP2
	-256FC-Output
LIF-SNN-5 [3]	Input
	-128C3-BN-MP2
	-128C3-BN-MP2
	-128C3-BN-MP2
	-128C3-BN-MP2
	-128C3-BN-MP2
	-512FC-AP10-Output
Res-SNN-18 [2]	Input
	-64C7-BN-MP3
	-64C3-BN-64C3-BN-64C3-BN-128C3-BN
	-128C3-BN-128C3-BN-128C3-BN-256C3-BN
	-256C3-BN-256C3-BN-256C3-BN-512C3-BN
	-512C3-BN-512C3-BN-512C3-BN-512C3-BN
	AdaptiveAP-512FC-Output

Table S4: Details of LIF-SNN and Res-SNN-18 network structures.

Hyper-parameter	Gesture	Gait-day	Gait-night	DailyAction-DVS
Max Epoch	200	200	150	200
Train Batch Size	32	32	32	32
Test Batch Size	4	4	4	4
Learning Rate	1e-4	1e-4	1e-4	1e-3
Threshold V_{th}	0.3	0.3	0.3	1
Decay factor β	0.3	0.3	0.3	0.5
Reset potential V_{reset}	0	0	0	0
Penalty coefficient λ	1e-8	1e-8	1e-8	1e-8
Dropout rate before FC	0	0.2	0	0
Dropout rate after LIF	0.5	0.5	0	0
Reduction factor r	2	2	2	2

Table S5: Hyper-parameter setting in Gesture, Gait-day, Gait-night, and DailyAction-DVS.

identical lighting and camera location, each subject carried out each motion. The duration of each recording is within 6s.

HAR-DVS[16]. The HARDVS dataset is collected with a DVS346 camera whose resolution is 346×260. HAR-DVS contains a total of 107,646 event streams and 300 classes of common human activities.

S2.2. Experimental Setup

In this work, we employ three network structures, including three-layer LIF-SNN[21], five-layer LIF-SNN[3], and Res-SNN-18[2]. The network structures of these baseline models are given in Table S4. We use the Adam optimizer to accelerate the training process and employ some standard training techniques of deep learning, such as batch normalization, dropout, etc. The corresponding hyper-parameters are shown in Table S5 and Table S6.

Hyper-parameter	HAR-DVS
Max Epoch	300
Train Batch Size	128
Test Batch Size	32
Learning Rate	1e-4
Threshold V_{th}	1
Decay factor β	0.5
Reset potential V_{reset}	0
Penalty coefficient λ	1e-8
Reduction factor r	2
DataAug	EventMix & TrivialAugment [13]

Table S6: Hyper-parameter setting in HAR-DVS.

S3. Ablation Study on ASA Module Design

In this section, we give some design details of the ASA module. In Step 1 of the ASA module (see Fig.3**b** in the main text), temporal-channel features are aggregated by using both average-pooling and max-pooling operations, which infer two different tensors $\boldsymbol{F}_{avg}, \boldsymbol{F}_{max} \in \mathbb{R}^{T \times c_n \times 1 \times 1}$. We get the importance map $\boldsymbol{M}$ by

$$\boldsymbol{M}' = \frac{1}{2} \otimes (\boldsymbol{F}_{avg} + \boldsymbol{F}_{max}) + \alpha \otimes \boldsymbol{F}_{avg} + \gamma \otimes \boldsymbol{F}_{max}, \quad \text{(S5)}$$

$$\boldsymbol{M} = \sigma\left(\boldsymbol{W}_2^n(\mathrm{ReLU}(\boldsymbol{W}_1^n(\boldsymbol{M}')))\right), \quad \text{(S6)}$$

where α and γ are trainable parameters which are initialised with 0.5, σ means the sigmoid function, $\boldsymbol{W}_1^n \in \mathbb{R}^{\frac{T}{r} \times T}$, $\boldsymbol{W}_2^n \in \mathbb{R}^{T \times \frac{T}{r}}$, and r represents the dimension reduction factor. $\boldsymbol{M}', \boldsymbol{M} \in \mathbb{R}^{T \times c_n \times 1 \times 1}$, and we share $\boldsymbol{W}_1^n$ and $\boldsymbol{W}_2^n$ on the channel dimension.

It should be noted that there are some critical design details here, including Eq. S5, Eq. S6, and the setting of reduction factor r. Eq. S5 and Eq. S6 were born out of the most classic attention network in CNNs, Squeeze-and-Excitation networks [5], which first proposed the concept of channel attention. In this work, Eq. S5 corresponds to the *squeeze* operation, which produces a temporal-channel descriptor by aggregating features across their spatial dimensions. Then, the *excitation* operation is followed, i.e., Eq. S6, which takes the squeezed information as input and produces attention scores.

Squeeze operation comes in a variety of methods. Here we examine the impact of some typical variants on recognition results. For Eq. S5, the original function in [5] is

$$\boldsymbol{M}' = \boldsymbol{F}_{avg}, \quad \text{(S7)}$$

and the following Convolutional Block Attention Module (CBAM) [19] exploits both average-pooling and max-pooling operations as

$$\boldsymbol{M}' = \frac{1}{2} \otimes (\boldsymbol{F}_{avg} + \boldsymbol{F}_{max}). \quad \text{(S8)}$$

Moreover, the Eq. S5 used in this work is inspired by the Hybrid Attention Module in [9]. We present the task accuracies corresponding to these three squeeze designs in Table S7. Results demonstrate that the best task performance is obtained using Eq. S5.

Design of $\boldsymbol{M}'$	Eq. S5 [9]	Eq. S7 [5]	Eq. S8 [19]
Acc.(%)	95.2	93.4	94.8

Table S7: Ablation study on squeeze operation. Based on three-layer SNN baseline [21], we perform ablation experiments on Gesture with $dt = 15, T = 60$.

Excitation operation is closely related to the number of additional parameters, the amount of computation, and the accuracy. The vanilla excitation operation was a two-layer fully connected neural network. Subsequent works [15, 20] attempt to optimize from the perspective of reducing the amount of computation and parameters. In this paper, after obtaining the attention matrix $\boldsymbol{M}'$, we did not use these scores to optimize the membrane potential in order to reduce the amount of additional computations. Therefore, accurately estimating which channels are important has a great influence on the final result. In Table S8, we report the effect of using different excitation operations, where the vanilla two-layer MLP works best.

Design of $\boldsymbol{M}$	Eq. S6 [5]	C1D [15]	SimAM [20]
Acc.(%)	95.2	93.8	92.7

Table S8: Ablation study on excitation operation. Based on three-layer SNN baseline [21], we perform ablation experiments on Gesture with $dt = 15, T = 60$.

Reduction factor r has a great impact on the accuracy. We assess the impact of r in Table S9. Considering the accuracy and the additional complexity brought by r (see Table S1), we uniformly set $r = 2$ in this paper.

r	1	2	4
Acc.(%)	95.2	95.2	94.4

Table S9: Ablation study on reduction factor r. Based on three-layer SNN baseline [21], we perform ablation experiments on Gesture with $dt = 15, T = 60$.

References

[1] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, et al. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7243–7252, 2017.

[2] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Thirty-fifth Conference on Neural Information Processing Systems (NIPS2021)*, 2021.

[3] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothee Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2661–2671, October 2021.

[4] 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.

[5] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[6] Yifan Hu, Yujie Wu, Lei Deng, and Guoqi Li. Advancing residual learning towards powerful deep spiking neural networks. *arXiv preprint arXiv:2112.08954*, 2021.

[7] Souvik Kundu, Gourav Datta, Massoud Pedram, and Peter A Beerel. Spike-thrift: Towards energy-efficient deep spiking neural networks by limiting spiking activity via attention-guided compression. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3953–3962, 2021.

[8] Souvik Kundu, Massoud Pedram, and Peter A Beerel. Hire-snn: Harnessing the inherent robustness of energy-efficient deep spiking neural networks by training with crafted input noise. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5209–5218, 2021.

[9] Guoqiang Li, Qi Fang, Linlin Zha, Xin Gao, and Nenggan Zheng. Ham: Hybrid attention module in deep convolutional neural networks for image classification. *Pattern Recognition*, 129:108785, 2022.

[10] Qianhui Liu, Dong Xing, Huajin Tang, De Ma, and Gang Pan. Event-based action recognition using motion information and spiking neural networks. In *IJCAI*, pages 1743–1749, 2021.

[11] Xin Liu, Mingyu Yan, Lei Deng, Yujie Wu, De Han, Guoqi Li, Xiaochun Ye, and Dongrui Fan. General spiking neural network framework for the learning trajectory from a noisy mmwave radar. *Neuromorphic Computing and Engineering*, 2(3):034013, 2022.

[12] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *International Conference on Learning Representations*, 2017.

[13] Samuel G Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 774–782, 2021.

[14] Priyadarshini Panda, Sai Aparna Aketi, and Kaushik Roy. Toward scalable, efficient, and accurate deep spiking neural networks with backward residual connections, stochastic softmax, and hybridization. *Frontiers in Neuroscience*, 14:653, 2020.

[15] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[16] Xiao Wang, Zongzhen Wu, Bo Jiang, Zhimin Bao, Lin Zhu, Guoqi Li, Yaowei Wang, and Yonghong Tian. Hardvs: Revisiting human activity recognition with dynamic vision sensors. *arXiv preprint arXiv:2211.09648*, 2022.

[17] Yanxiang Wang, Bowen Du, Yiran Shen, Kai Wu, Guangrong Zhao, Jianguo Sun, and Hongkai Wen. Ev-gait: Event-based robust gait recognition using dynamic vision sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6358–6367, 2019.

[18] Y. Wang, X. Zhang, Y. Shen, B. Du, G. Zhao, L. C. Cui Lizhen, and H. Wen. Event-stream representation for human gaits identification using deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[19] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[20] Lingxiao Yang, Ru-Yuan Zhang, Lida Li, and Xiaohua Xie. Simam: A simple, parameter-free attention module for convolutional neural networks. In *International Conference on Machine Learning*, pages 11863–11874. PMLR, 2021.

[21] Man Yao, Huanhuan Gao, Guangshe Zhao, Dingheng Wang, Yihan Lin, Zhaoxu Yang, and Guoqi Li. Temporal-wise attention spiking neural networks for event streams classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10221–10230, October 2021.

[22] Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, Bo Xu, and Guoqi Li. Attention spiking neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–18, 2023.

[23] Bojian Yin, Federico Corradi, and Sander M Bohté. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence*, 3(10):905–913, 2021.