S1 - More on Energy Consumption Analysis

This is the continuation of Section 3.4. In ANN models, Floating Point Operations (FLOPs) are frequently used as the metric to measure model's computational expenses. However, in the case of SNN models, the cost associates with additional parameters: the timestep T and the spike fire rate R. More specifically, timestep T represents the time-domain information in SNN, and spike fire rate R represents the ratio of the number of spikes to the total number of elements in a matrix [Kim *et al.*, 2022; Lee *et al.*, 2020]. Therefore, the computational expense of SNNs, referred to as Spiking Operations (SOPs), is expressed by the following equation:

$$SOPs(l) = T \times R \times FLOPs(l), \tag{S1}$$

where l is the layer of the architecture layer.

In our OST model, two basic layers are involved: the Convolutional (Conv) layer and the Linear layer. The FLOPs of the l-th Conv layer in ANNs are calculated as:

$$FL_C = 2k^2 \cdot h_{out} \cdot w_{out} \cdot c_{in} \cdot c_{out}, \tag{S2}$$

where k is the kernel size, (h_{out}, w_{out}) is the height-width vector of the output feature map, c_{in} and c_{out} are the input and output channel numbers, respectively. The FLOPs of the l-th Linear layer in ANNs are:

$$FL_L = d_{in} \cdot d_{out}, \tag{S3}$$

where d_{in} and d_{out} are the input and output dimensions of the Linear layer, respectively. Following Equation (S1), the cost of Conv layer and Linear layer can be represented as Equation (S4) and Equation (S5):

$$SOPs_C(l) = T \times R \times FL_C, \tag{S4}$$

$$SOPs_L(l) = T \times R \times FL_L. \tag{S5}$$

The spike firing rate is the proportion of non-zero elements within the spike tensor. Table S1 presents the spike firing rates for all spiking tensors in our OST-8-512 model.

Following the convention [Zhou *et al.*, 2023; Yao *et al.*, 2023; Kundu *et al.*, 2021; Panda *et al.*, 2020; Yin *et al.*, 2021], we assume the calculation for all operations are 32-bit floating point implementation on a 45nm chip. Therefore, the energy consumption of floating-point Multiply-and-Accumulate (MAC) and Accumulate (AC) are $E_{MAC} = 4.6pJ$ and $E_{AC} = 0.9pJ$, respectively. For identical operators (Conv and Linear), SNNs are theoretically more energy-efficient than their ANN counterparts, given that $E_{AC} \times T \times R < E_{MAC}$. As illustrated in Table S1, the spike firing rate (R) is exceedingly low for most spiking tensors, suggesting that the OST model is capable of achieving significantly reduced energy consumption.

	Matrix	Spike Firing Rate
SPS*	Conv1	0.21081×4
	Conv2	0.09489×4
	Conv3	0.06113×4
	Conv4	0.07868×4
Block1	$Q_{\mathcal{S}}$	0.37859
	$K_{\mathcal{S}}$	0.27312
	$X_{\mathcal{S}1}$	0.02492
	$G_{\mathcal{S}}$	0.05702
	$X_{\mathcal{S}2}$	0.03293
Block2	$Q_{\mathcal{S}}$	0.24378
	$K_{\mathcal{S}}$	0.15537
	$X_{\mathcal{S}1}$	0.03681
	$G_{\mathcal{S}}$	0.05408
	$X_{\mathcal{S}2}$	0.05102
Block3	$Q_{\mathcal{S}}$	0.38560
	$K_{\mathcal{S}}$	0.10754
	$X_{\mathcal{S}1}$	0.05788
	$G_{\mathcal{S}}$	0.03983
	$X_{\mathcal{S}2}$	0.02347
Block4	$Q_{\mathcal{S}}$	0.65051
	$K_{\mathcal{S}}$	0.07510
	$X_{\mathcal{S}1}$	0.04186
	$G_{\mathcal{S}}$	0.04793
	$X_{\mathcal{S}2}$	0.03565
Block5	$Q_{\mathcal{S}}$	0.82733
	$K_{\mathcal{S}}$	0.04249
	$X_{\mathcal{S}1}$	0.08343
	$G_{\mathcal{S}}$	0.05246
	$X_{\mathcal{S}2}$	0.07712
Block6	$Q_{\mathcal{S}}$	0.78324
	$K_{\mathcal{S}}$	0.03165
	$X_{\mathcal{S}1}$	0.04527
	$G_{\mathcal{S}}$	0.05183
	$X_{\mathcal{S}2}$	0.07617
Block7	$Q_{\mathcal{S}}$	0.67235
	$K_{\mathcal{S}}$	0.08474
	$X_{\mathcal{S}1}$	0.08181
	$G_{\mathcal{S}}$	0.06816
	$X_{\mathcal{S}2}$	0.12761
Block8	$Q_{\mathcal{S}}$	0.79305
	$K_{\mathcal{S}}$	0.04823
	$X_{\mathcal{S}1}$	0.05337
	$G_{\mathcal{S}}$	0.18035
	$X_{\mathcal{S}2}$	0.13328

Table S1: Spike firing rate of all spiking tensors in OST-8-512. The symbol * denotes that SPS using four timesteps in calculation, whereas Block1 to Block8 use only one timestep.

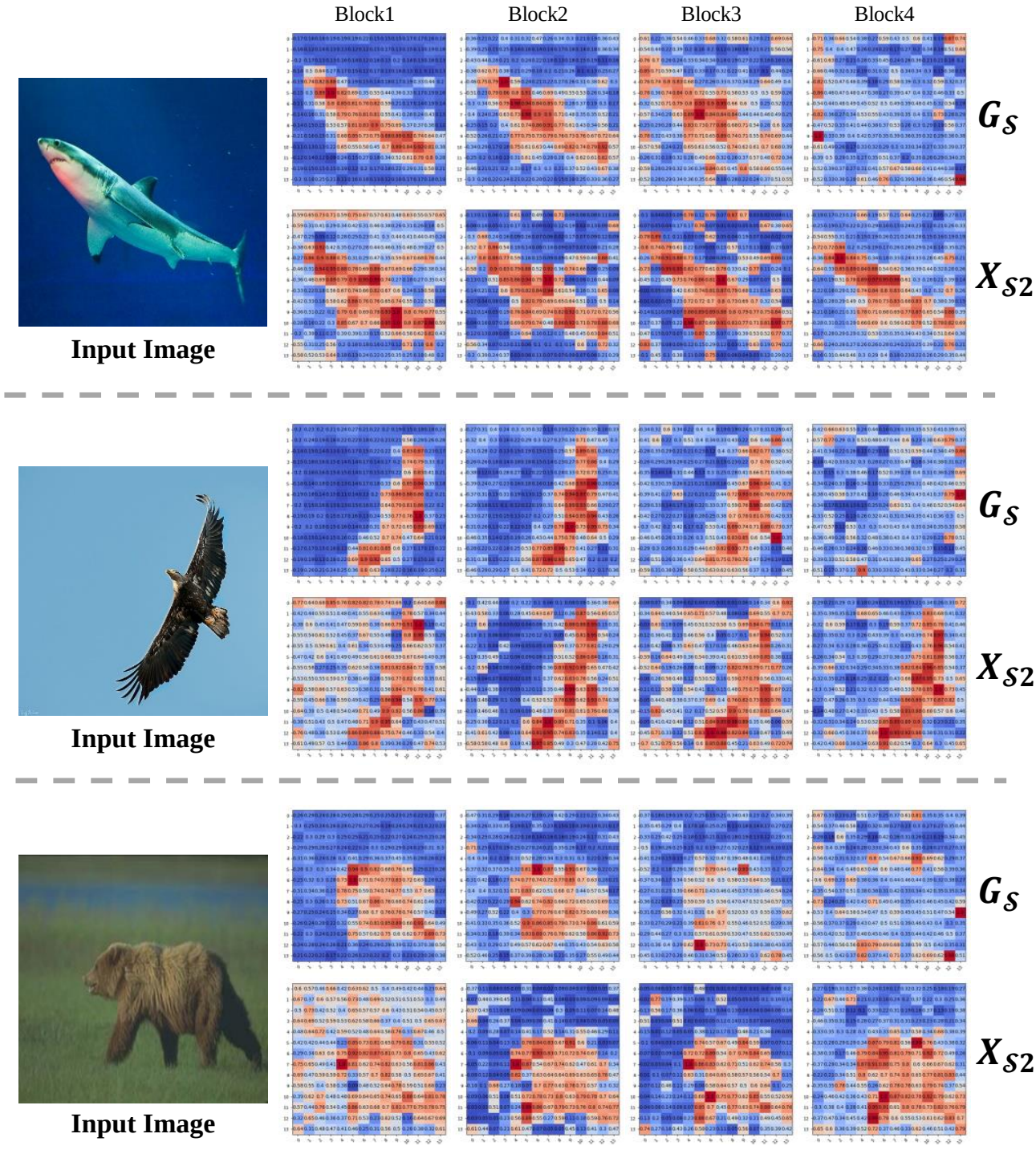

Figure S1: Attention map of the spiking gate mechanism based on spike firing rate. G_S is the gate tensor. X_S is the output derived from the spiking gate mechanism. The gate tensor G_S selects important information to obtain X_{S2}. The spatial resolution of each attention map is 14×14. Notably, the redder the color, the greater the attention; the bluer the color, the lesser the attention.

S2 - Spike Gating Mechanism

This is the supplementary for Section 4.1. **Spiking Gate Feed-forward (SGFF)** is proposed to replace MLP in Spikformer [Zhou *et al.*, 2023]. The implementation of SGFF is presented as Equation (S6) and Equation (S7):

$$G_{\mathcal{S}} = LIF(X_{\mathcal{S}}W^{G}), P_{\mathcal{F}} = X_{\mathcal{S}}W^{P}, \tag{S6}$$

$$SGFF(G_{\mathcal{S}}, P_{\mathcal{F}}) = LIF((G_{\mathcal{S}} \odot P_{\mathcal{F}})W^{O}), \tag{S7}$$

where W^{G} and $W^{P} \in \mathbb{R}^{D \times D^{''}}$, and $W^{O} \in \mathbb{R}^{D^{''} \times D}$. In addition, $D^{''}$ is the hidden dimension of *SGFF* and $\odot$ is the Element-wise product.

The spike gating mechanism, denoted as $G_{\mathcal{S}} \odot P_{\mathcal{F}}$, possesses the capability to selectively learn and control the flow of information. As shown in Table S1, the minimal value of the spike firing rate of $G_{\mathcal{S}}$ is usually below 0.1. This indicates that only less than 10% of the information passes through the spike gate. To enable a more intuitive view of the gate mechanism, the output results are visualized in Figure S1. In the case of each block (e.g. Block1, Block2, Block3, and Block4), the spike gate tensor $G_{\mathcal{S}}$ did select the crucial information, directing the attention of $X_{\mathcal{S}2}$ towards the target, either a shark, an eagle or a bear.

S3 - Experiment on Different MLP Ratios

Architecture	MLP Ratio	Params (M)	Time step	CIFAR10 Acc. (%)	CIFAR100 Acc. (%)
SDT-4-384	4	9.36	4	95.45	78.34
SDT-2-512	4	9.76	4	95.60	78.40
Spikformer-4-256	4	4.15	4	93.94	75.96
Spikformer-2-384	4	5.76	4	94.80	76.95
Spikformer-4-384	4	9.32	4	95.19	77.86
OST-4-256	2	**3.57**	1*	95.04	76.59
	3	4.38	1*	95.16	76.82
	4	5.15	1*	95.13	77.05
OST-2-384	2	5.88	1*	95.63	77.11
	3	6.77	1*	95.44	77.49
	4	7.62	1*	95.47	77.47
OST-4-384	2	7.82	1*	95.68	78.25
	3	9.63	1*	95.51	78.20
	4	11.37	1*	95.64	**78.76**
OST-2-512	2	10.29	1*	95.68	78.47
	3	11.87	1*	95.63	78.60
	4	13.45	1*	**95.83**	78.53

Table S2: The performance of OST with different MLP ratio and parameters on 'CIFAR10' and 'CIFAR100' datasets.

This section supplements the results of Section 4.1. Compared to Spikformer [Zhou *et al.*, 2023] and SDT [Yao *et al.*, 2023], our OST has a larger number of parameters, because OST incorporates three linear layers to realize the SGFF component, whereas both Spikformer and SDT utilize two linear layers for their MLP components. If we set the MLP ratio to 4, which is the same as Spikformer and SDT, OST would exhibit even larger parameters. Consequently, we use smaller MLP ratios (2 and 3) in our experiments. The results are presented in Table S2. We can see that OST attains superior performance with fewer parameters. More specifically, OST-4-256 with MLP ratio 2 has the fewest parameters (3.57M), but with a comparable performance as Spikformer-2-384, which has 5.76M parameters. Moreover, OST-4-384 with MLP ratio 2 has 7.82M parameters and achieves the highest accuracy on CIFAR10, whereas its accuracy on CIFAR100 is close to that of SDT-4-384 with 9.36M parameters.

S4 - Visualizing DVS128 Gesture Classification

Following Section 4.2, here we provide a more detailed visualization on DVS128 Gesture. Figure S2 illustrates some cases where SDT, Spikformer, and OST all failed to recognize the true label. Due to the significant variability in the input event stream across different timesteps (as illustrated in the left side of Figure S2 from ① to ⑥), all three models fail to capture the essential pattern of the event. Figure S3 shows a failure case in SDT, whereas OST was successful. The heat map of SDT focuses more on the right corner area, where there is not much movement happening during the event. On the contrary, the heat map of OST focuses on the left area, where the variation happens drastically along with the gesture. We further analyse a case which Spikformer failed but OST succeeded. Compared to Spikformer, OST also exhibits a more concentrated pattern of spiking activation over the movement of the gesture.

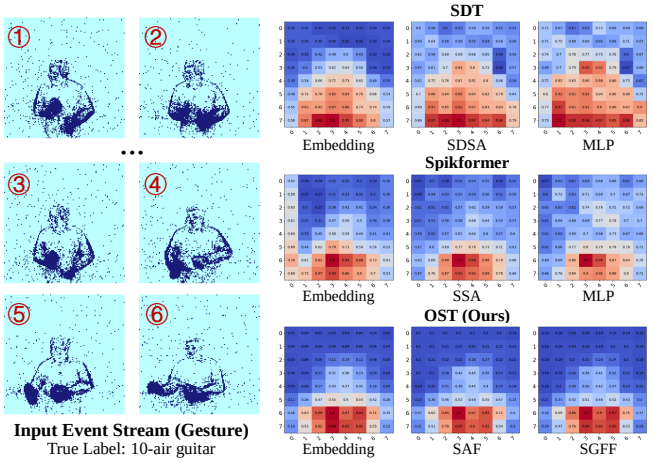

Figure S2: The failure case of SDT (SDSA and MLP), Spikformer (SSA and MLP), and OST (with *SAF* and *SGFF*) on Gesture dataset.

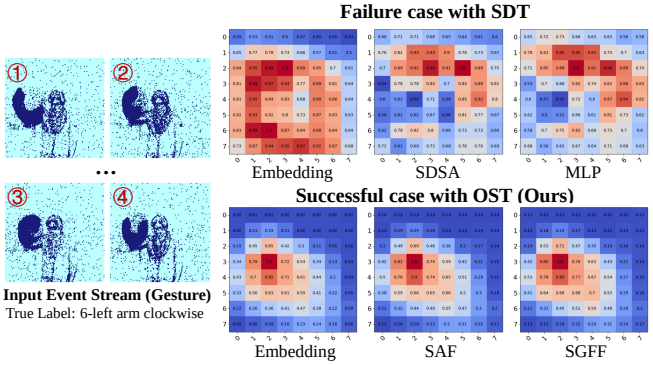

Figure S3: Heat map examples of SDT and OST on Gesture dataset.

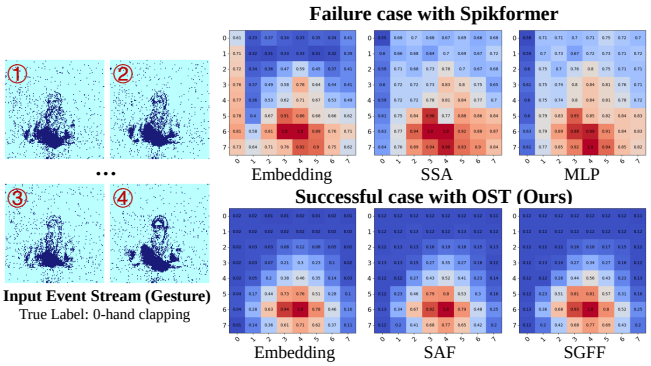

Figure S4: Heat map examples of Spikformer and OST on Gesture dataset.

S5 - Impact of Initial Timesteps, Static Dataset

As mentioned in Section 4.3, the initial timesteps does have impact, affecting the training time, inference latency, and energy consumption of the SNN model. For a direct training of SNN models, e.g. SDT, Spikformer, and OST, the training time builds up significantly along with the increase of the number of time steps. Therefore, we experiment with different timesteps, e.g. 8, 4, 2, and 1, not only focusing on accuracy but also on the average training time in one epoch (denoted as average runtime). The experimental settings are as follows: batch size 128, mixed precision set as true, using the same single NVIDIA 3090. The results are presented in Table S3, which shows that overall, OST can achieve better performance with less runtime. More specifically, OST achieves 76.96% when the initial timestep is set to 1, which is 2.9% and 1.44% higher than Spikformer and SDT. With the initial timesteps 2, the accuracy of OST is 78.11%, gaining 2.25% and 1.76% improvement compared to Spikformer and SDT. The average runtime of OST with initial timesteps 1 and 2 is similar or less than those of Spikformer and SDT. Moreover, as the initial timesteps increases to 4 and 8, the average runtime of OST is much less than those of Spikformer and SDT. For an initial timesteps 4, OST achieves the best accuracy of 78.76%, using only 75% and 68% average runtime compared to Spikformer and SDT. For initial timesteps 8, the performance of OST is similar to SDT, but with only 62% of SDT's average runtime. A further observation is that our OST is less sensitive to the drop of the initial timesteps.

Architecture	Initial Timesteps	CIFAR100 Acc. (%)	Average Runtime (s)
SDT-4-384	1	75.52	24
	2	76.35	34
	4	78.34	59
	8	79.02	114
Spikformer-4-384	1	74.06	25
	2	75.86	37
	4	77.86	65
	8	78.70	126
OST-4-384 (Ours)	1	76.96	24
	2	78.11	31
	4	78.76	44
	8	79.01	71

Table S3: The performance and average runtime under different initial timesteps on 'CIFAR100' dataset.

S6 - Impact of Initial Timesteps, Neuro. Dataset

Following the experiments in Section S5, this section focuses on Neuromorphic images, which are constructed using Dynamic Vision Sensor (DVS), containing local events, e.g. changes in brightness. The timesteps here, reflecting the magnitude of changes caused by events, are more critical for neuromorphic image classification than that for static image classification. Therefore, to further validate the effectiveness of the proposed OST, we employ bigger initial timesteps (e.g. 16, 10, 8, 4, and 2) in our experiments. The results are shown in Table S4. We can see that OST achieved better performance, especially on smaller initial timesteps, e.g. 10, 8, 4, and 2. Similar to that in Section S5, the performance of SDT is sensitive to the initial timesteps, as shown in the first line of the top block in Table S4. The performance of SDT drops dramatically by 11.5% and 8.9% on DVS128 Gesture and CIFAR10-DVS, respectively. This indicates that the high-performance SDT model relies on the lengthy timesteps. For Spikformer (the middle block), the performance drop is relatively less, 9.8% and 5.9% on DVS128 Gesture and CIFAR10-DVS. Still, that is a severe drop compared to that of OST, 4.9% and 3.1% respectively on two datasets.

Architecture	Initial Timesteps	Gesture Acc. (%)	CIFAR10-DVS Acc. (%)
SDT-2-256	2	87.8 (-11.5%)	71.1 (-8.9%)
	4	90.3 (-9.0%)	71.9 (-8.1%)
	8	92.4 (-6.9%)	74.6 (-5.4%)
	10	94.8 (-4.5%)	75.7 (-4.3%)
	16	**99.3** (0.0%)	80.0 (0.0%)
Spikformer-2-256	2	88.5 (-9.8%)	75.0 (-5.9%)
	4	92.7 (-5.6%)	76.8 (-4.1%)
	8	94.8 (-3.5%)	77.3 (-3.6%)
	10	96.9 (-1.4%)	78.9 (-2.0%)
	16	98.3 (0.0%)	<u>80.9</u> (0.0%)
OST-2-256 (Ours)	2	94.1 (-4.9%)	78.1 (-3.1%)
	4	95.1 (-3.9%)	78.3 (-2.9%)
	8	96.1 (-2.9%)	79.5 (-1.7%)
	10	97.2 (-1.8%)	79.6 (-1.6%)
	16	<u>99.0</u> (0.0%)	**81.2** (0.0%)

Table S4: The performance of SDT, Spikformer, and OST (Ours) with different timestep on neuromorphic datasets.

Summary of Section S5 and Section S6 The experimental results presented in Sections S5 and S6 validate the following advantages of OST: (1) faster running times, especially noticeable for lengthy timesteps ($T \geq 4$); (2) better performance, particularly with a small timesteps ($T < 4$); (3) less sensitive to the reduction of timesteps. These advantages of our proposed OST hold true on both static image classifications as well as neuromorphic image classifications.

References

[Kim *et al.*, 2022] Youngeun Kim, Hyoungseob Park, Abhishek Moitra, Abhiroop Bhattacharjee, Yeshwanth Venkatesha, and Priyadarshini Panda. Rate coding or direct coding: Which one is better for accurate, robust, and energy-efficient spiking neural networks? In *Proc, of ICASSP*, pages 71–75. IEEE, 2022.

[Kundu *et al.*, 2021] 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 *Proc. of CVPR*, pages 5209–5218, 2021.

[Lee *et al.*, 2020] Chankyu Lee, Syed Shakib Sarwar, Priyadarshini Panda, Gopalakrishnan Srinivasan, and Kaushik Roy. Enabling spike-based backpropagation for training deep neural network architectures. *Frontiers in neuroscience*, page 119, 2020.

[Panda *et al.*, 2020] 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.

[Yao *et al.*, 2023] Man Yao, JiaKui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo XU, and Guoqi Li. Spike-driven transformer. In *Proc. of NeurIPS*, 2023.

[Yin *et al.*, 2021] 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.

[Zhou *et al.*, 2023] Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng YAN, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *Proc. of ICLR*, 2023.