Spike-driven Transformer

Man Yao[1,2,3]*, **Jiakui Hu**[1,4*], **Zhaokun Zhou**[3,4*], **Li Yuan**[3,4],
Yonghong Tian[3,4], **Bo Xu**[1], **Guoqi Li**[1†]

[1]Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]School of Automation Science and Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi, China
[3]Peng Cheng Laboratory, Shenzhen, Guangzhou, China
[4]Peking University, Beijing, China

Abstract

Spiking Neural Networks (SNNs) provide an energy-efficient deep learning option due to their unique spike-based event-driven (i.e., spike-driven) paradigm. In this paper, we incorporate the spike-driven paradigm into Transformer by the proposed Spike-driven Transformer with *four* unique properties: i) Event-driven, no calculation is triggered when the input of Transformer is zero; ii) Binary spike communication, all matrix multiplications associated with the spike matrix can be transformed into sparse additions; iii) Self-attention with linear complexity at both token and channel dimensions; iv) The operations between spike-form Query, Key, and Value are mask and addition. Together, there are only sparse addition operations in the Spike-driven Transformer. To this end, we design a novel Spike-Driven Self-Attention (SDSA), which exploits only mask and addition operations without any multiplication, and thus having up to $87.2\times$ lower computation energy than vanilla self-attention. Especially in SDSA, the matrix multiplication between Query, Key, and Value is designed as the mask operation. In addition, we rearrange all residual connections in the vanilla Transformer before the activation functions to ensure that all neurons transmit binary spike signals. It is shown that the Spike-driven Transformer can achieve 77.1% top-1 accuracy on ImageNet-1K, which is the state-of-the-art result in the SNN field. The source code is available at Spike-driven Transfromer.

1 Introduction

One of the most crucial computational characteristics of bio-inspired Spiking Neural Networks (SNNs) [1] is spike-based event-driven (spike-driven): i) When a computation is event-driven, it is triggered sparsely as events (spike with address information) occur; ii) If only binary spikes (0 or 1) are employed for communication between spiking neurons, the network's operations are synaptic ACcumulate (AC). When implementing SNNs on neuromorphic chips, such as TrueNorth [2], Loihi [3], and Tianjic [4], only a small fraction of spiking neurons at any moment being active and the rest being idle. Thus, spike-driven neuromorphic computing that only performs sparse addition operations is regarded as a promising low-power alternative to traditional AI [5–7].

Although SNNs have apparent advantages in bio-plausibility and energy efficiency, their applications are limited by poor task accuracy. Transformers have shown high performance in various tasks for their self-attention [8–10]. Incorporating the effectiveness of Transformer with the high energy efficiency of SNNs is a natural and exciting idea. There has been some research in this direction, but

*Equal contribution. manyao@stu.xjtu.edu.cn; jiakuihu29@gmail.com; zhouzhaokun@stu.pku.edu.cn
†Corresponding author, guoqi.li@ia.ac.cn

Preprint. Under review.

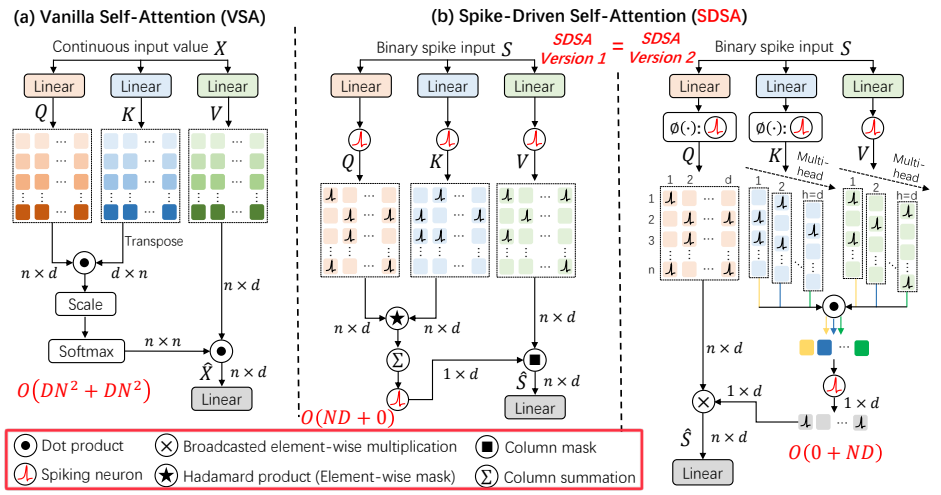

Figure 1: Comparison Vanilla Self-Attention (VSA) and our Spike-Driven Self-Attention (SDSA). **(a)** is a typical Vanilla Self-Attention (VSA) [8]. **(b)** are two equivalent versions of SDSA. The input of SDSA are binary spikes. In SDSA, there are only mask and sparse additions. Version 1: Spike Q and K first perform element-wise mask, i.e., Hadamard product; then column summation and spike neuron layer are adopted to obtain the binary attention vector; finally, the binary attention vector is applied to the spike V to mask some channels (features). Version 2: An equivalent version of Version 1 (see Section 3.3) reveals that SDSA is a unique type of linear attention (spiking neuron layer is the kernel function) whose time complexity is linear with both token and channel dimensions. Typically, performing self-attention in VSA and SDSA requires $2N^2D$ multiply-and-accumulate and $0.02ND$ accumulate operations respectively, where N is the number of tokens, D is the channel dimensions, 0.02 is the non-zero ratio of the matrix after the mask of Q and K. Thus, the self-attention operator between the spike Q, K, and V has almost no energy consumption.

all so far have relied on "hybrid computing". Namely, Multiply-and-ACcumulate (MAC) operations dominated by vanilla Transformer components and AC operations caused by spiking neurons both exist in the existing spiking Transformers. One popular approach is to replace some of the neurons in Transformer with spiking neurons to do a various of tasks [11–22], and keeping the MAC-required operations like dot-product, softmax, scale, etc.

Though hybrid computing helps reduce the accuracy loss brought on by adding spiking neurons to the Transformer, it can be challenging to benefit from SNN's low energy cost, especially given that current spiking Transformers are hardly usable on neuromorphic chips. To address this issue, we propose a novel Spike-driven Transformer that achieves the spike-driven nature of SNNs throughout the network while having great task performance. Two core modules of Transformer, Vanilla Self-Attention (VSA) and Multi-Layer Perceptron (MLP), are re-designed to have a spike-driven paradigm.

The three input matrices for VSA are Query (Q), Key (K), and Value (V), (Fig. 1(a)). Q and K first perform similarity calculations to obtain the attention map, which includes three steps of matrix multiplication, scale and softmax. The attention map is then used to weight the V (another matrix multiplication). The typical spiking self-attentions in the current spiking Transformers [20, 19] would convert Q, K, V into spike-form before performing two matrix multiplications similar to those in VSA. The distinction is that spike matrix multiplications can be converted into addition, and softmax is not necessary [20]. But these methods not only yield large integers in the output (thus requiring an additional scale multiplication for normalization to avoid gradient vanishing), but also fails to exploit the full energy-efficiency potential of the spike-driven paradigm combined with self-attention.

We propose Spike-Driven Self-Attention (SDSA) to address these issues, including two aspects (see SDSA Version 1 in Fig. 1(b)): i) Hadamard product replaces matrix multiplication; ii) matrix column-wise summation and spiking neuron layer take the role of softmax and scale. The former can be considered as not consuming energy because the Hadamard product between spikes is equivalent to the element-wise mask. The latter also consumes almost no energy since the matrix to be summed column-by-column is very sparse (typically, the ratio of non-zero elements is less than 0.02). We also

observe that SDSA is a special kind of linear attention [23, 24], i.e., Version 2 of Fig. 1(b). In this view, the spiking neuron layer that converts Q, K, and V into spike form is a kernel function.

Additionally, existing spiking Transformers [20, 12] usually follow the SEW-SNN residual design [25], whose shortcut is spike addition and thus outputs multi-bit (integer) spikes. This shortcut can satisfy event-driven, but introduces integer multiplication. We modified the residual connections throughout the Transformer architecture as shortcuts between membrane potentials [26, 27] to address this issue (Section 3.2). The proposed Spike-driven Transformer improves task accuracy on both static and neuromorphic event-based datasets. The main contributions of this paper are as follows:

- We propose a novel Spike-driven Transformer that only exploits sparse addition. This is the first time that the spike-driven paradigm has been incorporated into Transformer, and the proposed model is hardware-friendly to neuromorphic chips.
- We design a Spike-Driven Self-Attention (SDSA). The self-attention operator between spike Query, Key, Value is replaced by mask and sparse addition with essentially no energy consumption. SDSA is computationally linear in both tokens and channels. Overall, the energy cost of SDSA (including Query, Key, Value generation parts) is 87.2× lower than its vanilla self-attention counterpart.
- We rearrange the residual connections so that all spiking neurons in Spike-driven Transformer communicate via binary spikes.
- Extensive experiments show that the proposed architecture can outperform or comparable to State-Of-The-Art (SOTA) SNNs on both static and neuromorphic datasets. We achieved 77.1% accuracy on ImageNet-1K, which is the SOTA result in the SNN field.

2 Related Works

Bio-inspired Spiking Neural Networks can profit from advanced deep learning and neuroscience knowledge concurrently [28, 5, 7, 29]. Many biological mechanisms are leveraged to inspire SNN's neuron modeling [1, 30], learning rules [31, 32], etc. Existing studies have shown that SNNs are more suited for incorporating with brain mechanisms, e.g., long short-term memory [33, 34], attention [35, 27], etc. Moreover, while keeping its own spike-driven benefits, SNNS have greatly improved its task accuracy by integrating deep learning technologies like network architecture [26, 25], gradient backpropagation [36, 37], normalization [38, 39], etc. Our goal is to combine SNN and Transformer architectures. One way is to discretize Transformer into spike form through neuron equivalence [40, 41], i.e., ANN2SNN, but this requires a long simulation timestep and boosts the energy consumption. We employ the direct training method, using the first SNN layer as the spike encoding layer and applying surrogate gradient training [42].

Neuromorphic Chips. As opposed to the compute and memory separated processors used in ANNs, neuromorphic chips use non-von Neumann architectures, which are inspired by the structure and function of the brain [5, 7, 28]. Because of the choice that uses spiking neurons and synapses as basic units, neuromorphic chips [43–45, 2, 46, 3, 4] have unique features, such as highly parallel operation, collocated processing and memory, inherent scalability, and spike-driven computing, etc. Typical neuromorphic chips consume tens to hundreds of mWs [47]. Conv and MLP in neuromorphic chips are equivalent to a cheap addressing algorithm [48] since the sparse spike-driven computing, i.e., to find out which synapses and neurons need to be involved in the addition operation. Our Transformer design strictly follows the spike-driven paradigm, thus it is friendly to deploy on neuromorphic chips.

Efficient Transformers. Transformer and its variants have been widely used in numerous tasks, such as natural language processing [8, 49, 50] and computer vision [51–53]. However, deploying these models on mobile devices with limited resources remains challenging because of their inherent complexity [24, 54]. Typical optimization methods include convolution and self-attention mixing [54, 55], Transformer's own mechanism (token mechanism [56, 57], self-attention [58, 59], multi-head [60, 61] and so on) optimization, etc. An important direction for efficient Transformers is linear attention on tokens since the computation scale of self-attention is quadratic with token number. Removing the softmax in self-attention and re-arranging the computation order of Query, Key, and Value is the main way to achieve linear attention [59, 62, 23, 63, 64, 61, 65]. For the spiking Transformer, softmax cannot exist, thus spiking Transformer can be a kind of linear attention.

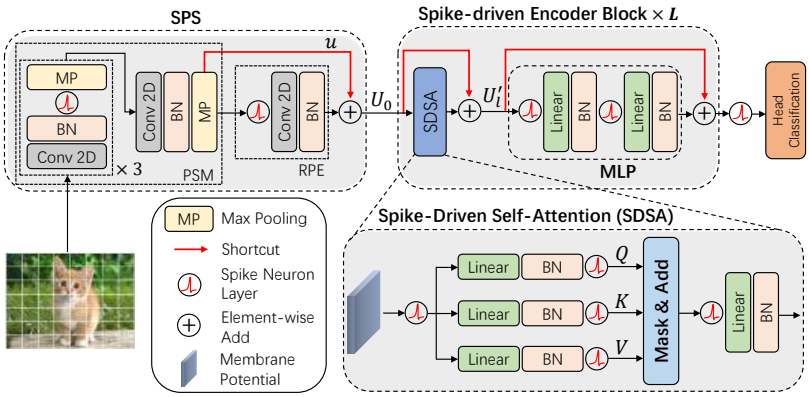

Figure 2: The overview of Spike-driven Transformer. We follow the network structure in [20], but make two new key designs. First, we propose a Spike-driven Self-Attention (SDSA) module, which consists of only mask and sparse addition operations (Fig. 1(b)). Second, we redesign the shortcuts in the whole network, involving position embedding, self-attention, and MLP parts. As indicated by the red line, the shortcut is constructed before the spike neuron layer. That is, we establish residual connections between membrane potentials to make sure that the values in the spike matrix are all binary, which allows the multiplication of the spike and weight matrices to be converted into addition operations. By contrast, previous works [20, 19] build shortcut between spike tensors in different layers, resulting in the output of spike neurons as multi-bit (integer) spikes.

3 Spike-driven Transformer

We propose a Spike-driven Transformer, which incorporates Transformer into the spike-driven paradigm with only sparse addition. We first briefly introduce the spike neuron layer, then introduce the overview and components of the Spike-driven Transformer one by one.

The spiking neuron model is simplified from the biological neuron model [66, 30]. Leaky Integrate-and-Fire (LIF) spiking neuron [1], which have biological neuronal dynamics and are easy to simulate on a computer, is uniformly adopted in this work. The dynamics of the LIF layer [36] is governed by

$$U[t] = H[t-1] + X[t], \tag{1}$$
$$S[t] = \mathrm{Hea}\left(U[t] - u_{th}\right), \tag{2}$$
$$H[t] = V_{reset}S[t] + \left(\beta U[t]\right)\left(\mathbf{1} - S[t]\right), \tag{3}$$

where t denote the timestep, $U[t]$ means the membrane potential which is produced by coupling the spatial input information $X[t]$ and the temporal input $H[t-1]$, where $X[t]$ can be obtained through operators such as Conv, MLP, and self-attention. When membrane potential exceeds the threshold u_{th}, the neuron will fire a spike, otherwise it will not. Thus, the spatial output tensor $S[t]$ contains only 1 or 0. $\mathrm{Hea}(\cdot)$ is a Heaviside step function that satisfies $\mathrm{Hea}\left(x\right) = 1$ when $x \geq 0$, otherwise $\mathrm{Hea}\left(x\right) = 0$. $H[t]$ indicates the temporal output, where V_{reset} denotes the reset potential which is set after activating the output spiking. $\beta < 1$ is the decay factor, if the spiking neuron does not fire, the membrane potential $U[t]$ will decay to $H[t]$.

3.1 Overall Architecture

Fig. 2 shows the overview of Spike-driven Transformer that includes four parts: Spiking Patch Splitting (SPS), SDSA, MLP, and a linear classification head. For the SPS part, we follow the design in [20]. Given a 2D image sequence $I \in \mathbb{R}^{T\times C\times H\times W}$, the Patch Splitting Module (PSM), i.e., the first four Conv layers, linearly projects and splits it into a sequence of N flattened spike patches s with D dimensional channel, where T (images are repeated T times in the static dataset as input), C, H, and W denote timestep, channel, height and width of the 2D image sequence. Another Conv layer

is then used to generate Relative Position Embedding (RPE). Together, the SPS part is written as:

$$u = \mathrm{PSM}\left(I\right), \qquad I \in \mathbb{R}^{T\times C\times H\times W}, u \in \mathbb{R}^{T\times N\times D} \tag{4}$$

$$s = \mathcal{SN}(u), \qquad s \in \mathbb{R}^{T\times N\times D} \tag{5}$$

$$\mathrm{RPE} = \mathrm{BN}(\mathrm{Conv2d}(s)), \qquad \mathrm{RPE} \in \mathbb{R}^{T\times N\times D} \tag{6}$$

$$U_0 = u + \mathrm{RPE}, \qquad U_0 \in \mathbb{R}^{T\times N\times D} \tag{7}$$

where u and U_0 are the output membrane potential tensor of PSM and SPS respectively, $\mathcal{SN}(\cdot)$ denote the spike neuron layer. Then we pass the U_0 to the L-block Spike-driven Transformer encoder, which consists of a SDSA and a MLP block. Residual connections are applied to membrane potentials in both SDSA and MLP block. SDSA provides an efficient approach to model the local-global information of images utilizing spike Q, K, and V without scale and softmax (see Sec. 3.3). A Global Average-Pooling (GAP) is utilized on the processed feature from spike-driven encoder and outputs the D-dimension channel which will be sent to the fully-connected-layer Classification Head (CH) to output the prediction Y. The three parts SDSA, MLP and CH can be written as follows:

$$S_0 = \mathcal{SN}(U_0), \qquad S_0 \in \mathbb{R}^{T\times N\times D} \tag{8}$$

$$U_l^{'} = \mathrm{SDSA}(S_{l-1}) + U_{l-1}, \qquad U_l^{'} \in \mathbb{R}^{T\times N\times D}, l = 1...L \tag{9}$$

$$S_l^{'} = \mathcal{SN}(U_l^{'}), \qquad S_l^{'} \in \mathbb{R}^{T\times N\times D}, l = 1...L \tag{10}$$

$$S_l = \mathcal{SN}(\mathrm{MLP}(S_l^{'}) + U_l^{'}), \qquad S_l \in \mathbb{R}^{T\times N\times D}, l = 1...L \tag{11}$$

$$Y = \mathrm{CH}(\mathrm{GAP}(S_L)), \tag{12}$$

where $U_l^{'}$ and $S_l^{'}$ represent membrane potential and spike tensor output by SDSA at l-th layer.

3.2 Membrane Shortcut in Spike-driven Transformer

Residual connection [67, 68] is a crucial basic operation in Transformer architecture. There are three shortcut techniques in existing Conv-based SNNs [27]. Vanilla Res-SNN [38], similar to vanilla Res-CNN [67], performs a shortcut between membrane potential and spike. Spike-Element-Wise (SEW) Res-SNN [25] employs a shortcut to connect the output spikes in different layers. Membrane Shortcut (MS) Res-SNN [26], creating a shortcut between membrane potential of spiking neurons in various layers. There is no uniformly standard shortcut in the current SNN community, and SEW shortcut is adopted by existing spiking Transformers [20, 12]. As shown in Eq. 7, Eq. 9 and Eq. 11, we leverage the membrane shortcut in the proposed Spike-driven Transformer for four reasons:

- *Spike-driven* refers to the ability to transform matrix multiplication between weight and spike tensors into sparse additions. Only binary spikes can support the spike-driven function. However, the values in the spike tensors are multi-bit (integer) spikes, as the SEW shortcut builds the addition between binary spikes. By contrast, as shown in Eq. 8, Eq. 10, Eq. 11, $\mathcal{SN}$ is followed by the MS shortcut, which ensures that there are always only binary spike signals in the spike tensor.
- *High performance*. The task accuracy of MS-Res-SNN is higher than that of SEW-Res-SNN [26, 27, 25], also in Transformer-based SNN (see Table 6 in this work).
- *Bio-plausibility*. MS shortcut can be understood as an approach to optimize the membrane potential distribution. This is consistent with other neuroscience-inspired methods to optimize the internal dynamics of SNNs, such as complex spiking neuron design [69], attention mechanism [27], long short-term memory [34], recurrent connection [70], information maximization [71], etc.
- *Dynamical isometry*. MS-Res-SNN has been proven [26] to satisfy dynamical isometry theory [72], which is a theoretical explanation of well-behaved deep neural networks [73, 74].

3.3 Spike-driven Self-Attention

Spike-Driven Self-Attention (SDSA) Version 1. As shown in Fig. 2(b) left part, given a spike input feature sequence $S \in \mathbb{R}^{T\times N\times D}$, float-point Q, K, and V in $\mathbb{R}^{T\times N\times D}$ are calculated by three learnable linear matrices, respectively. Note, the linear operation here is only addition, because the input S is a spike tensor. A spike neuron layer $\mathcal{SN}(\cdot)$ follows, converting Q, K, V into spike tensor Q_S, K_S, and V_S. SDSA Version 1 (SDSA-V1) is presented as:

$$\mathrm{SDSA}(Q,K,V) = g(Q_S,K_S)\otimes V_S = \mathcal{SN}\left(\mathrm{SUM_c}\left(Q_S\otimes K_S\right)\right)\otimes V_S, \tag{13}$$

where $\otimes$ is the Hadamard product, $g(\cdot)$ is used to compute the attention map, $\mathrm{SUM_c}(\cdot)$ represents the sum of each column. The outputs of both $g(\cdot)$ and $\mathrm{SUM_c}(\cdot)$ are D-dimensional row vectors. The Hadamard product between spike tensors is equivalent to the mask operation.

Discussion on SDSA. Since the Hadamard product among Q_S, K_S, and V_S in $\mathbb{R}^{N\times D}$ (we here assume $T=1$ for mathematical understanding) can be exchanged, Eq. 13 can also be written as:

$$\mathrm{SDSA}(Q,K,V)=Q_S\otimes g(K_S,V_S)=Q_S\otimes\mathcal{SN}\left(\mathrm{SUM_c}\left(K_S\otimes V_S\right)\right). \tag{14}$$

In this view, Eq.14 is a linear attention [23, 62] whose computational complexity is linear in token number N because K_S and V_S can participate in calculate first. This is thanks to the softmax operation in the VSA is dropped here. The function of softmax needs to be replaced by the kernel function. Specific to our SDSA, $\mathcal{SN}(\cdot)$ is the kernel function. Further, we can assume a special case [61], $H=D$, i.e., the number of channels per head is one. After the self-attention operation is performed on the H heads respectively, the outputs are concatenated together. Specifically,

$$\mathrm{SDSA}(Q^i,K^i,V^i)=\mathcal{SN}(Q^i)g(K^i,V^i)=\mathcal{SN}(Q^i)\mathcal{SN}\left(\mathcal{SN}(Q^i)^{\mathrm{T}}\odot\mathcal{SN}(V^i)\right), \tag{15}$$

where Q^i,K^i,V^i in $\mathbb{R}^{N\times 1}$ are the i-th vectors in Q,K,V respectively, $\odot$ is the dot product operation. The output of $g(K^i,V^i)$ is a scalar, 0 or 1. Since the operation between $\mathcal{SN}(Q^i)$ and $g(K^i,V^i)$ is a mask, the whole SDSA only needs to be calculated $H=D$ times for $g(K^i,V^i)$. The computational complexity of SDSA is $O(0+ND)$, which is linear with both N and D (see Fig. 1(b) right part). Vectors K^i and V^i are very sparse, typically less than 0.01 (Table 2). Together, the whole SDSA only needs about $0.02ND$ times of addition, and its energy consumption is negligible.

Interestingly, Eq. 14 actually converts the soft vanilla self-attention to hard self-attention, where the attention scores in soft and hard attention are continuous- and binary-valued, respectively [75]. Thus, the practice of the spike-driven paradigm in this work leverages binary self-attention scores to directly mask unimportant channels in the sparse spike Value tensor. Although this introduces a slight loss of accuracy (Table 6), $\mathrm{SDSA}(\cdot)$ consumes almost no energy.

4 Theoretical Energy Consumption Analysis

Three key computational modules in deep learning are Conv, MLP, and self-attention. In this Section, We discuss how the spike-driven paradigm achieves high energy efficiency on these operators.

Spike-driven in Conv and MLP. Spike-driven combines two properties, event-driven and binary spike-based communication. The former means that no computation is triggered when the input is zero. The binary restriction in the latter indicates that there are only additions. In summary, in spike-driven Conv and MLP, matrix multiplication is transformed into sparse addition, which is implemented as addressable addition in neuromorphic chips [48].

Spike-driven in Self-attention. Q_S, K_S, V_S in spiking self-attention involve two matrix multiplications. One approach is to perform multiplication directly between Q_S, K_S, V_S, which is then converted to sparse addition, like spike-driven Conv and MLP. The previous work [20] did just that. We provide a new scheme that performs element-wise multiplication between Q_S, K_S, V_S. Since all elements in spike tensors are either 0 or 1, element multiplication is equivalent to a mask operation with no energy consumption. Mask operations can be implemented in neuromorphic chips through addressing algorithms [48] or AND logic operations [4].

Energy Consumption Comparison. The times of floating-point operations (FLOPs) is often used to estimate the computational burden in ANNs, where almost all FLOPs are MAC. Under the same architecture, the energy cost of SNN can be estimated by combining the spike firing rate R and simulation timestep T if the FLOPs of ANN is known. Table 1 shows the energy consumption of Conv, self-attention, and MLP modules of the same scale in vanilla and our Spike-driven Transformer.

5 Experiments

We evaluate our method on both static datasets ImageNet [76], CIFAR-10/100 [77], and neuromorphic datasets CIFAR10-DVS [78], DVS128 Gesture [79].

Experimental Setup on ImageNet. For the convenience of comparison, we generally continued the experimental setup in [20]. The input size is set to 224×224. The batch size is set to 128 or

Table 1: Energy evaluation. FL_{Conv} and FL_{MLP} represent the FLOPs of the Conv and MLP models in the ANNs, respectively. R_C, R_M, $\overline{R}$, $\widehat{R}$ denote the spike firing rates (the proportion of non-zero elements in the spike matrix) in various spike matrices. We give the strict definitions and calculation methods of these indicators in the Supplementary due to space constraints.

		Vanilla Transformer [51]	Spike-driven Transformer (**This work**)
SPS	First Conv	$E_{MAC} \cdot FL_{Conv}$	$E_{MAC} \cdot T \cdot R_C \cdot FL_{Conv}$
	Other Conv	$E_{MAC} \cdot FL_{Conv}$	$E_{AC} \cdot T \cdot R_C \cdot FL_{Conv}$
Self-attention	Q, K, V	$E_{MAC} \cdot 3ND^2$	$E_{AC} \cdot T \cdot \overline{R} \cdot 3ND^2$
	$f(Q, K, V)$	$E_{MAC} \cdot 2N^2D$	$E_{AC} \cdot T \cdot \widehat{R} \cdot ND$
	Scale	$E_M \cdot N^2$	-
	Softmax	$E_{MAC} \cdot 2N^2$	-
	Linear	$E_{MAC} \cdot FL_{MLP0}$	$E_{AC} \cdot T \cdot R_{M0} \cdot FL_{MLP0}$
MLP	Layer 1	$E_{MAC} \cdot FL_{MLP1}$	$E_{AC} \cdot T \cdot R_{M1} \cdot FL_{MLP1}$
	Layer 2	$E_{MAC} \cdot FL_{MLP2}$	$E_{AC} \cdot T \cdot R_{M2} \cdot FL_{MLP2}$

256 during 310 training epochs with a cosine-decay learning rate whose initial value is 0.0005. The optimizer is Lamb. The image is divided into $N = 196$ patches using the SPS module. Standard data augmentation techniques, like random augmentation, mixup, are also employed in training. Details of the training and experimental setup on ImageNet are given in the supplementary material.

Accuracy analysis on ImageNet. Our experimental results on ImageNet are given in Table 3. We first compare our model performance with the baseline spiking Transformer (i.e., SpikFormer [20]). The five network architectures consistent with those in SpikFormer are adopted by this work. We can see that under the same parameters, our accuracies are significantly better than the corresponding baseline models. For instance, the Spike-driven Transformer-8-384 is 2.0% higher than SpikFormer-8-384. It is worth noting that the Spike-driven Transformer-8-768 obtains 76.32% (input 224×224) with 66.34M, which is 1.5% higher than the corresponding SpikFormer. We further expand the inference resolution to 288×288, obtaining 77.1%, which is the SOTA result of the SNN field on ImageNet.

Table 2: Spike Firing Rate (SFR) of Spike-driven Self-attention in 8-512. Average SFR is the mean of SFR over $T = 4$, and 8 SDSA blocks.

SDSA	Average SFR
Q_S	0.0091
K_S	0.0090
$g(Q_S, K_S)$	0.0713
V_S	0.1649
Output of $\mathrm{SDSA}(\cdot)$, $\hat{V}_S$	0.0209

We then compare our results with existing Res-SNNs. Whether it is vanilla Res-SNN [38], MS-Res-SNN [26] or SEW-Res-SNN [25], the accuracy of Spike-driven Transformer-8-768 (77.1%) is the highest. Att-MS-Res-SNN [27] also achieves 77.1% accuracy by plugging an additional attention auxiliary module [80, 81] in MS-Res-SNN, but it destroys the spike-driven nature and requires more parameters (78.37M vs. 66.34M) and training time (1000epoch vs. 310epoch). Furthermore, the proposed Spike-driven Transformer outperforms by more than 72% at various network scales, while Res-SNNs have lower performance with a similar amount of parameters. For example, **Spike-driven Transformer-6-512 (This work)** vs. SEW-Res-SNN-34 vs. MS-Res-SNN-34: Param, **23.27M** vs. 21.79M vs. 21.80M; Acc, **74.11%** vs. 67.04% vs. 69.15%.

Power analysis on ImageNet. Compared with prior works, the Spike-driven Transformer shines in energy cost (Table 3). We first make an intuitive comparison of energy consumption in the SNN field. **Spike-driven Transformer-8-512 (This work)** vs. SEW-Res-SNN-50 vs. MS-Res-SNN-34: Power, **4.50mJ** vs. 4.89mJ vs. 5.11mJ; Acc, **74.57%** vs. 67.78% vs. 69.42%. That is, our model achieves +6.79% and +5.15% accuracy higher than previous SEW and MS Res-SNN backbones with lower energy consumption. What is more attractive is that the energy efficiency of the Spike-driven Transformer will be further expanded as the model scale grows because its computational complexity is linear in both token and channel dimensions. For instance, in an 8-layer network, as the channel dimension increases from **384** to **512** and **768**, SpikFormer [20] has **1.98**×(7.73mJ/3.90mJ), **2.57**×(11.57mJ/4.50mJ), and **3.52**×(21.47mJ/6.09mJ) higher energy consumption than our Spike-

Table 3: Evaluation on ImageNet. Power is the average theoretical energy consumption when predicting an image from the test set. The power data in this work is evaluated according to Table 1, and data for other works were obtained from related papers. Spiking Transformer-L-D represents a model with L encoder blocks and D channels. *The input crops are enlarged to 288×288 in inference. The default inference input resolution for other models is 224×224.

Methods	Architecture	Spike -driven	Param (M)	Power (mJ)	Time Step	Acc
Hybrid training [82]	ResNet-34	✓	21.79	-	250	61.48
TET [83]	SEW-ResNet-34	✗	21.79	-	4	68.00
Spiking ResNet [84]	ResNet-50	✓	25.56	70.93	350	72.75
tdBN [38]	Spiking-ResNet-34	✗	21.79	6.39	6	63.72
SEW ResNet [25]	SEW-ResNet-34	✗	21.79	4.04	4	67.04
	SEW-ResNet-50	✗	25.56	4.89	4	67.78
	SEW-ResNet-101	✗	44.55	8.91	4	68.76
	SEW-ResNet-152	✗	60.19	12.89	4	69.26
MS ResNet [26]	MS-ResNet-18	✓	11.69	4.29	4	63.10
	MS-ResNet-34	✓	21.80	5.11	4	69.42
	MS-ResNet-104*	✓	77.28	10.19	4	76.02
Att MS ResNet [27]	Att-MS-ResNet-18	✗	11.87	0.48	1	63.97
	Att-MS-ResNet-34	✗	22.12	0.57	1	69.15
	Att-MS-ResNet-104*	✗	78.37	7.30	4	**77.08**
ResNet	Res-CNN-104	✗	77.28	54.21	1	76.87
Transformer	Transformer-8-512	✗	29.68	41.77	1	**80.80**
Spikformer [20]	Spiking Transformer-8-384	✗	16.81	7.73	4	70.24
	Spiking Transformer-6-512	✗	23.37	9.41	4	72.46
	Spiking Transformer-8-512	✗	29.68	11.57	4	73.38
	Spiking Transformer-10-512	✗	36.01	13.89	4	73.68
	Spiking Transformer-8-768	✗	66.34	21.47	4	74.81
Spike-driven Transformer (Ours)	Spiking Transformer-8-384	✓	16.81	3.90	4	72.28
	Spiking Transformer-6-512	✓	23.37	3.56	4	74.11
	Spiking Transformer-8-512	✓	29.68	**1.13**	1	71.68
	Spiking Transformer-8-512	✓	29.68	4.50	4	74.57
	Spiking Transformer-10-512	✓	36.01	5.53	4	74.66
	Spiking Transformer-8-768*	✓	66.34	6.09	4	**77.07**

driven Transformer. At the same time, our task performance on these three network structures has improved by **+2.0%**, **+1.2%**, and **+1.5%**, respectively.

Then we compare the energy cost between Spike-driven and ANN Transformer. Under the same structure, such as 8-512, the power required by the ANN-ViT (41.77mJ) is **9.3**× that of the spike-driven counterpart (4.50mJ). Further, the energy advantage will extend to **36.7**× if we set $T = 1$ in the Spike-driven version (1.13mJ). Although the accuracy of $T = 1$ (here we are direct training) will be lower than $T = 4$, it can be compensated by special training methods [85] in future work.

Table 4: Energy Consumption of Self-attention. E_1 and E_2 (including energy consumption to generate Q, K, V) represent the power of self-attention mechanism in ANN and spike-driven.

Models	E_1 (pJ)	E_2 (pJ)	E_1/E_2
8-384	6.7e8	1.6e7	**42.6**
8-512	1.2e9	2.1e7	**57.2**
8-768	2.7e9	3.1e7	**87.2**

Sparse spike firing is the key for Spike-driven Transformer to achieve high energy efficiency. As shown in Table 2, the Spike Firing Rate (SFR) of the self-attention part is very low, where the SFR of Q_S and Q_K are both less than 0.01. Since the mask (Hadamard product) operation does not consume energy, the number of additions required by the $\mathrm{SUM_c}\left(Q_S \otimes K_S\right)$ is less than $0.02ND$ times. The operation between the vector output by $g(Q_S, K_S)$ and V_S is still a column mask that does not consume energy. Consequently, in the whole self-attention part, the energy consumption of spike-driven self-attention can be lower than $87.2\times$ of ANN self-attention (see Table 4).

Table 5: Experimental Results on CIFAR10/100, DVS128 Gesture and CIFAR10-DVS.

Methods	Spike-driven	CIFAR10-DVS		DVS128 Gesture		CIFAR-10		CIFAR-100	
		T	Acc	T	Acc	T	Acc	T	Acc
tdBN [38]	✗	10	67.8	40	96.9	6	93.2	-	-
PLIF [86]	✓	20	74.8	20	97.6	8	93.5	-	-
Dspike [87]	✗	10	75.4	-	-	6	94.3	6	74.2
DSR [88]	✓	10	77.3	-	-	20	95.4	20	**78.5**
Spikformer [20]	✗	16	**80.9**	16	98.3	4	95.5	4	78.2
DIET-SNN[89]	✗	-	-	-	-	5	92.7	5	69.67
ANN(ResNet19)	✗	-	-	-	-	1	94.97	1	75.4
ANN(Transformer4-384)	✗	-	-	-	-	1	96.7	1	81.02
This Work	✓	16	80.0	16	**99.3**	4	**95.6**	4	78.4

Table 6: Studies on Spiking Transformer-2-512.

Model	CIFAR-10	CIFAR-100
Baseline [20]	93.12	73.17
+ SDSA	93.09 (-0.03)	72.83 (-0.34)
+ MS	93.93 (+0.81)	74.63 (+1.46)
This work	93.82 (+0.73)	74.41 (+1.24)

Experimental results on CIFAR-10/100, CIFAR10-DVS, and DVS128 Gesture are conducted in Table 5. These four datasets are relatively small compared to ImageNet. CIFAR-10/100 are static image classification datasets. Gesture and CIFAR10-DVS are neuromorphic action classification datasets, which need to convert the event stream into frame sequences before processing. DVS128 Gesture is a gesture recognition dataset. CIFAR10-DVS is a neuromorphic dataset converted from CIFAR-10 by shifting image samples to be captured by the DVS camera. We basically keep the experimental setup in [20], including the network structure, training settings, etc., and details are given in the supplementary material. As shown in Table 5, we achieve SOTA results on Gesture (99.3%) and CIFAR-10 (95.6%), and comparable results to SOTA on other datasets.

Ablation study. To implement the spike-driven paradigm in Transformer, we design a new SDSA module and reposition the residual connections in the entire network based on the SpikFormer [20]. We organize ablation studies on CIFAR10/100 to analyze their impact. Results are given in Table 6. We adopt spiking Transformer-2-512 as the baseline structure. It can be observed that SDSA incurs a slight performance loss. As discussed in Section 3.3, SDSA actually masks some unimportant channels directly. In Fig. 3, we plot the attention maps (the detailed drawing method is given in the supplementary material), and we can observe: i) SDSA can optimize intermediate features, such as masking background information; ii) SDSA greatly reduces the spike firing rate of $\hat{V}_S$, thereby reducing energy cost. On the other hand, the membrane shortcut leads to significant accuracy improvements, consistent with the experience of Conv-based MS-SNN [26, 25]. Comprehensively, the proposed Spike-driven Transformer simultaneously achieves better accuracy and lower energy consumption (Table 3).

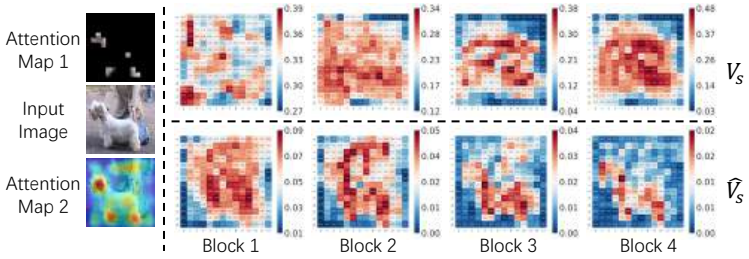

Figure 3: Attention Map Based on Spike Firing Rate (SFR). V_S is the Value tensor. $\hat{V}_S$ is the output of $\mathrm{SDSA}(\cdot)$. The spike-driven self-attention mechanism masks unimportant channels in V_S to obtain $\hat{V}_S$. Each pixel on V_S and $\hat{V}_S$ represents the SFR at a patch. The spatial resolution of each attention map is 14×14 (196 patches). The redder the higher the SFR, the bluer the smaller the SFR.

6 Conclusion

We propose a Spike-driven Transformer that combines the low power of SNN and the excellent accuracy of the Transformer. There is only sparse addition in the proposed Spike-driven Transformer. To this end, we design a novel Spike-Driven Self-Attention (SDSA) module and rearrange the

location of residual connections throughout the network. The complex and energy-intensive matrix multiplication, softmax, and scale in the vanilla self-attention are dropped. Instead, we employ mask, addition, and spike neuron layer to realize the function of the self-attention mechanism. Moreover, SDSA has linear complexity with both token and channel dimensions. Extensive experiments are conducted on static image and neuromorphic datasets, verifying the effectiveness and efficiency of the proposed method. We hope our investigations pave the way for further research on Transformer-based SNNs and inspire the design of next-generation neuromorphic chips.

References

[1] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.

[2] Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014.

[3] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018.

[4] Jing Pei, Lei Deng, et al. Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature*, 572(7767):106–111, 2019.

[5] Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019.

[6] 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.

[7] Catherine D Schuman, Shruti R Kulkarni, Maryam Parsa, J Parker Mitchell, Bill Kay, et al. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, 2(1):10–19, 2022.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[9] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 45(1):87–110, 2022.

[10] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.

[11] Yudong Li, Yunlin Lei, and Xu Yang. Spikeformer: A novel architecture for training high-performance low-latency spiking neural network. *arXiv preprint arXiv:2211.10686*, 2022.

[12] Nathan Leroux, Jan Finkbeiner, and Emre Neftci. Online transformers with spiking neurons for fast prosthetic hand control. *arXiv preprint arXiv:2303.11860*, 2023.

[13] Jiqing Zhang, Bo Dong, Haiwei Zhang, Jianchuan Ding, Felix Heide, Baocai Yin, and Xin Yang. Spiking transformers for event-based single object tracking. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 8801–8810, 2022.

[14] Jiyuan Zhang, Lulu Tang, Zhaofei Yu, Jiwen Lu, and Tiejun Huang. Spike transformer: Monocular depth estimation for spiking camera. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 34–52. Springer, 2022.

[15] Zhengkun Tian, Jiangyan Yi, Jianhua Tao, Ye Bai, Shuai Zhang, and Zhengqi Wen. Spike-Triggered Non-Autoregressive Transformer for End-to-End Speech Recognition. In *Proc. Interspeech 2020*, pages 5026–5030, 2020.

[16] Rui-Jie Zhu, Qihang Zhao, and Jason K Eshraghian. Spikegpt: Generative pre-trained language model with spiking neural networks. *arXiv preprint arXiv:2302.13939*, 2023.

[17] Etienne Mueller, Viktor Studenyak, Daniel Auge, and Alois Knoll. Spiking transformer networks: A rate coded approach for processing sequential data. In *2021 7th International Conference on Systems and Informatics (ICSAI)*, pages 1–5. IEEE, 2021.

[18] Minglun Han, Qingyu Wang, Tielin Zhang, Yi Wang, Duzhen Zhang, and Bo Xu. Complex dynamic neurons improved spiking transformer network for efficient automatic speech recognition. *Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI 2023)*, 2023.

[19] Shihao Zou, Yuxuan Mu, Xinxin Zuo, Sen Wang, and Li Cheng. Event-based human pose tracking by spiking spatiotemporal transformer. *arXiv preprint arXiv:2303.09681*, 2023.

[20] 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.

[21] Guangyao Chen, Peixi Peng, Guoqi Li, and Yonghong Tian. Training full spike neural networks via auxiliary accumulation pathway. *arXiv preprint arXiv:2301.11929*, 2023.

[22] Chenlin Zhou, Liutao Yu, Zhaokun Zhou, Han Zhang, Zhengyu Ma, Huihui Zhou, and Yonghong Tian. Spikingformer: Spike-driven residual learning for transformer-based spiking neural network. *arXiv preprint arXiv:2304.11954*, 2023.

[23] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[24] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28, 2022.

[25] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34:21056–21069, 2021.

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

[27] Man Yao, Guangshe Zhao, Hengyu Zhang, Hu Yifan, Lei Deng, Yonghong Tian, Bo Xu, and Guoqi Li. Attention spiking neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–18, 01 2023.

[28] Guoqi Li, Lei Deng, Huajin Tang, Gang Pan, Yonghong Tian, Kaushik Roy, and Wolfgang Maass. Brain inspired computing: A systematic survey and future trends. 2023.

[29] Man Yao, Yuhong Chou, Guangshe Zhao, Xiawu Zheng, Yonghong Tian, Bo Xu, and Guoqi Li. Probabilistic modeling: Proving the lottery ticket hypothesis in spiking neural network. *arXiv preprint arXiv:2305.12148*, 2023.

[30] Eugene M Izhikevich. Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572, 2003.

[31] Tielin Zhang, Xiang Cheng, Shuncheng Jia, Mu-ming Poo, Yi Zeng, and Bo Xu. Self-backpropagation of synaptic modifications elevates the efficiency of spiking and artificial neural networks. *Science Advances*, 7(43):eabh0146, 2021.

[32] Yujie Wu, Rong Zhao, Jun Zhu, Feng Chen, Mingkun Xu, Guoqi Li, Sen Song, Lei Deng, Guanrui Wang, Hao Zheng, et al. Brain-inspired global-local learning incorporated with neuromorphic computing. *Nature Communications*, 13(1):65, 2022.

[33] Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. *Advances in Neural Information Processing Systems*, 31, 2018.

[34] Arjun Rao, Philipp Plank, Andreas Wild, and Wolfgang Maass. A long short-term memory for ai applications in spike-based neuromorphic hardware. *Nature Machine Intelligence*, 4(5):467–479, 2022.

[35] 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*, pages 10221–10230, 2021.

[36] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience*, 12:331, 2018.

[37] 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.

[38] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11062–11070, 2021.

[39] Lei Deng, Yujie Wu, Yifan Hu, Ling Liang, Guoqi Li, Xing Hu, Yufei Ding, Peng Li, and Yuan Xie. Comprehensive snn compression using admm optimization and activity regularization. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[40] Shikuang Deng and Shi Gu. Optimal conversion of conventional artificial neural networks to spiking neural networks. In *International Conference on Learning Representations*, 2021.

[41] Jibin Wu, Chenglin Xu, Xiao Han, Daquan Zhou, Malu Zhang, Haizhou Li, and Kay Chen Tan. Progressive tandem learning for pattern recognition with deep spiking neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7824–7840, 2021.

[42] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 1311–1318, 2019.

[43] Johannes Schemmel, Daniel Brüderle, Andreas Grübl, Matthias Hock, Karlheinz Meier, and Sebastian Millner. A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1947–1950. IEEE, 2010.

[44] Eustace Painkras, Luis A Plana, Jim Garside, Steve Temple, Francesco Galluppi, Cameron Patterson, David R Lester, Andrew D Brown, and Steve B Furber. Spinnaker: A 1-w 18-core system-on-chip for massively-parallel neural network simulation. *IEEE Journal of Solid-state Circuits*, 48(8):1943–1953, 2013.

[45] Ben Varkey Benjamin, Peiran Gao, Emmett McQuinn, Swadesh Choudhary, Anand R Chandrasekaran, Jean-Marie Bussat, Rodrigo Alvarez-Icaza, John V Arthur, Paul A Merolla, and Kwabena Boahen. Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of the IEEE*, 102(5):699–716, 2014.

[46] Juncheng Shen, De Ma, Zonghua Gu, Ming Zhang, Xiaolei Zhu, Xiaoqiang Xu, Qi Xu, Yangjing Shen, and Gang Pan. Darwin: A neuromorphic hardware co-processor based on spiking neural networks. *Science China Information Sciences*, 59(2):1–5, 2016.

[47] Arindam Basu, Lei Deng, Charlotte Frenkel, and Xueyong Zhang. Spiking neural network integrated circuits: A review of trends and future directions. In *2022 IEEE Custom Integrated Circuits Conference (CICC)*, pages 1–8, 2022.

[48] Ole Richer, Ning Qiao, Qian Liu, and Sadique Sheik. Event-driven spiking convolutional neural network. *WIPO Patent*, page WO2020207982A1, 2020.

[49] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[50] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[51] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representa- tions (ICLR)*, 2020.

[52] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[53] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[54] Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 35:12934–12949, 2022.

[55] Haokui Zhang, Wenze Hu, and Xiaoyu Wang. Parc-net: Position aware circular convolution with merits from convnets and transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 613–630. Springer, 2022.

[56] Yongming Rao, Zuyan Liu, Wenliang Zhao, Jie Zhou, and Jiwen Lu. Dynamic spatial sparsification for efficient vision transformers and convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[57] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 558–567, 2021.

[58] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12259–12269, 2021.

[59] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

[60] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in Neural Information Processing Systems*, 32, 2019.

[61] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, and Judy Hoffman. Hydra attention: Efficient attention with many heads. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 35–49. Springer, 2023.

[62] Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosformer: Rethinking softmax in attention. In *International Conference on Learning Representations*, 2022.

[63] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3531–3539, 2021.

[64] Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah Smith, and Lingpeng Kong. Random feature attention. In *International Conference on Learning Representations*, 2021.

[65] Abdelrahman Shaker, Muhammad Maaz, Hanoona Rasheed, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Swiftformer: Efficient additive attention for transformer-based real-time mobile vision applications. *arXiv preprint arXiv:2303.15446*, 2023.

[66] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500–544, 1952.

[67] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[68] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 630–645, Cham, 2016. Springer International Publishing.

[69] Xingting Yao, Fanrong Li, Zitao Mo, and Jian Cheng. GLIF: A unified gated leaky integrate-and-fire neuron for spiking neural networks. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[70] 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.

[71] Yufei Guo, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Yinglei Wang, Xuhui Huang, and Zhe Ma. Im-loss: information maximization loss for spiking neural networks. *Advances in Neural Information Processing Systems*, 35:156–166, 2022.

[72] Zhaodong Chen, Lei Deng, Bangyan Wang, Guoqi Li, and Yuan Xie. A comprehensive and modularized statistical framework for gradient norm equality in deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):13–31, 2022.

[73] Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. In *International Conference on Machine Learning*, pages 5393–5402. PMLR, 2018.

[74] Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. Deepnet: Scaling transformers to 1,000 layers. *arXiv preprint arXiv:2203.00555*, 2022.

[75] Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, Zheng-Ning Liu, Peng-Tao Jiang, Tai-Jiang Mu, Song-Hai Zhang, Ralph R Martin, Ming-Ming Cheng, and Shi-Min Hu. Attention mechanisms in computer vision: A survey. *Computational Visual Media*, 8(3):331–368, 2022.

[76] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[77] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[78] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. Cifar10-dvs: an event-stream dataset for object classification. *Frontiers in Neuroscience*, 11:309, 2017.

[79] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, Jeff Kusnitz, Michael Debole, Steve Esser, Tobi Delbruck, Myron Flickner, and Dharmendra Modha. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7243–7252, 2017.

[80] 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.

[81] 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.

[82] Nitin Rathi, Gopalakrishnan Srinivasan, Priyadarshini Panda, and Kaushik Roy. Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation. *arXiv preprint arXiv:2005.01807*, 2020.

[83] Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting. In *International Conference on Learning Representations*, 2022.

[84] Yangfan Hu, Huajin Tang, and Gang Pan. Spiking deep residual networks. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–6, 2021.

[85] Sayeed Shafayet Chowdhury, Nitin Rathi, and Kaushik Roy. One timestep is all you need: training spiking neural networks with ultra low latency. *arXiv preprint arXiv:2110.05929*, 2021.

[86] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée 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, 2021.

[87] Yuhang Li, Yufei Guo, Shanghang Zhang, Shikuang Deng, Yongqing Hai, and Shi Gu. Differentiable Spike: Rethinking Gradient-Descent for Training Spiking Neural Networks. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, pages 23426–23439, 2021.

[88] Qingyan Meng, Mingqing Xiao, Shen Yan, Yisen Wang, Zhouchen Lin, and Zhi-Quan Luo. Training high-performance low-latency spiking neural networks by differentiation on spike representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12444–12453, 2022.

[89] Nitin Rathi and Kaushik Roy. Diet-snn: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–9, 2021.

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

[91] 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.

[92] 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.

[93] 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.

[94] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022.

[95] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

S1 Energy Consumption Analysis Details

We show the theoretical energy consumption estimation method of the proposed Spike-driven Transformer in Table 1 of the main text. Compared to the vanilla Transformer counterpart, the spiking version requires information on timesteps T and spike firing rates (R). Therefore, we only need to evaluate the FLOPs of the vanilla Transformer, and T and R are known, we can get the theoretical energy consumption of spike-driven Transformer.

The FLOPs of the n-th Conv layer in ANNs [90] are:

$$FL_{Conv} = (k_n)^2 \cdot h_n \cdot w_n \cdot c_{n-1} \cdot c_n, \tag{S1}$$

where k_n is the kernel size, (h_n, w_n) is the output feature map size, c_{n-1} and c_n are the input and output channel numbers, respectively. The FLOPs of the m-th MLP layer in ANNs are:

$$FL_{MLP} = i_m \cdot o_m, \tag{S2}$$

where i_m and o_m are the input and output dimensions of the MLP layer, respectively.

The spike firing rate is defined as the proportion of non-zero elements in the spike tensor. In Table S1, we present the spike firing rates for all spiking tensors in spike-driven Transformer-8-512. In addition, $\overline{R}$ in Table 1 indicates the average of the spike firing rates of Q_S, K_S, and V_S. $\widehat{R}$ is the sum of the spike firing rates of Q_S and K_S.

Refer to previous works[91, 70, 92, 27], we assume the data for various operations are 32-bit floating point implementation in 45nm technology [93], in which $E_{MAC} = 4.6pJ$ and $E_{AC} = 0.9pJ$. Overall, for the same operator (Conv, MLP, Self-attention), as long as $E_{AC} \times T \times R < E_{MAC}$, SNNs are theoretically more energy efficient than counterpart ANNs. $E_{AC} \times T$ is usually a constant, thus sparser spikes (smaller R) result in lower energy cost.

S2 Experiment Details

Datasets. We employ two types of datasets: static image classification and neuromorphic classification. The former includes ImageNet-1K [76], CIFAR-10/100 [77]. The latter contains CIFAR10-DVS [78] and DVS128 Gesture [79].

ImageNet-1K is the most typical static image dataset, which is widely used in the field of image classification. It offers a large-scale natural image dataset of 1.28 million training images and 50k test images, with a total of 1,000 categories. CIFAR10 and CIFAR100 are smaller datasets in image classification tasks that are often used for algorithm testing. The CIFAR-10 dataset consists of 60,000 images in 10 classes, with 6,000 images per class. The CIFAR-100 dataset has 60,000 images divided into 100 classes, each with 600 images.

CIFAR10-DVS is an event-based neuromorphic dataset converted from CIFAR10 by scanning each image with repeated closed-loop motion in front of a Dynamic Vision Sensor (DVS). There are a total of 10,000 samples in CIFAR10-DVS, with each sample lasting 300ms. The temporal and spatial resolutions are μs and 128×128, respectively. DVS128 Gesture is an event-based gesture recognition dataset, which has the temporal resolution in μs level and 128×128 spatial resolution. It records 1342 samples of 11 gestures, and each gesture has an average duration of 6 seconds.

Data Preprocessing. SNNs are a kind of spatio-temporal dynamic network that can naturally deal with temporal tasks. When working with static image classification datasets, it is common practice in the field to repeatedly input the same image at each timestep. As our results in Table 3 show, multiple timesteps lead to better accuracy, but also require more training time and computing hardware requirements, as well as greater inference energy consumption.

By contrast, neuromorphic datasets (i.e., event-based datasets) can fully exploit the energy-efficient advantages of SNNs with spatio-temporal dynamics. Specifically, neuromorphic datasets are produced by event-based (neuromorphic) cameras, such as DVS [94]. Compared with conventional cameras, DVS poses a new paradigm shift in visual information acquisition, which encode the time, location, and polarity of the brightness changes for each pixel into event streams with a μs level temporal resolution. Events (spike signals with address information) are generated only when the brightness of the visual scene changes. This fits naturally with the event-driven nature of SNNs. Only when there is an event input, some spiking neurons of SNNs will be triggered to participate in the computation.

Typically, event streams are preprocessed into frame sequences as input to SNNs. Details can be referred to previous work [35].

Experimental Steup. The experimental setup in this work generally follows [20]. The experimental settings of ImageNet-1K have been given in the main text. Here we mainly give the network settings on four small datasets. As shown in Table 5, we employ timesteps $T = 4$ on static CIFAR-10 and CIFAR-100, and $T = 16$ on neuromorphic CIFAR10-DVS and Gesture. The training epoch for these four datasets is 200. The batch size is 32 for CIFAR10/100, 16 for Gesture and CIFAR10-DVS. The learning rate is initialized to 0.0005 for CIFAR10/100, 0.0003 for Gesture, and 0.01 for CIFAR10-DVS. All of them are reduced with cosine decay. We follow [20] to apply data augmentation on Gesture and CIFAR10-DVS. In addition, the network structures used in CIFAR-10, CIFAR-100, CIFAR10-DVS, and Gesture are: spike-driven Transformer-2-512, spike-driven Transformer-2-512, spike-driven Transformer-2-256, spike-driven Transformer-2-256.

S3 Attention Map

Spike-Driven Self-Attention (SDSA). Here we first briefly review the proposed spike-driven self-attention. Given a single head spike input feature sequence $S \in \mathbb{R}^{T\times N\times D}$, float-point Q, K, and V in $\mathbb{R}^{T\times N\times D}$ are calculated by three learnable linear matrices, respectively. A spike neuron layer $\mathcal{SN}(\cdot)$ follows, converting float-point Q, K, V into spike tensor Q_S, K_S, and V_S. Spike-driven self-attention is presented as:

$$\hat{V}_S = \mathrm{SDSA}(Q, K, V) = g(Q_S, K_S) \otimes V_S = \mathcal{SN}\left(\mathrm{SUM_c}\left(Q_S \otimes K_S\right)\right) \otimes V_S, \tag{S3}$$

where $\otimes$ is the Hadamard product, $g(\cdot)$ is used to compute the attention map, $\mathrm{SUM_c}(\cdot)$ represents the sum of each column. The outputs of both $g(\cdot)$ and $\mathrm{SUM_c}(\cdot)$ are D-dimensional row vectors. The Hadamard product between spike tensors is equivalent to the mask operation. We denote the output of $\mathrm{SDSA}(Q, K, V)$ as $\hat{V}_S$.

Self-attention mechanism allows the model to capture long-range dependencies by attending to relevant parts of the input sequence regardless of the distance between them. In Eq. S3, SDSA adopts hard attention. The output of attention map $g(Q_S, K_S)$ is a vector containing only 0 or 1. Therefore, the whole spike-driven self-attention can be understood as masking unimportant channels in the Value tensor V_S. Note, instead of scale and softmax operations, we exploit Hadamard product, column element sum, and spiking neuron layer to generate binary attention scores. Q_S and K_S are very sparse (typically less than 0.01, see Table S1), the value of summing $Q_S \otimes K_S$ column by column does not fluctuate much, thus the scale operation is not needed here.

Attention Map. In a spike-driven self-attention layer, the V_S and $\hat{V}_S$ of T timesteps and H heads are averaged. The new V_S and $\hat{V}_S$ output is the spike firing rate, which we plot in Fig. S1. This allows us to observe how the attention score modulates spike firing.

Table S1: Spike Firing Rates in Spike-driven Transformer-8-512.

			$T=1$	$T=2$	$T=3$	$T=4$	Average
SPS		Conv1	0.0665	0.1260	0.1004	0.1451	0.1095
		Conv2	0.0465	0.0689	0.0597	0.0541	0.0573
		Conv3	0.0333	0.0453	0.0368	0.0394	0.0387
		Conv4	0.0948	0.1864	0.1792	0.1885	0.1622
Block 1	SDSA	Input	0.2873	0.3590	0.3630	0.3625	0.3430
		V_S	0.2629	0.3094	0.3011	0.3104	0.2959
		Q_S	0.0142	0.0202	0.0218	0.0219	0.0195
		K_S	0.0144	0.0227	0.0234	0.0246	0.0213
		$g(Q_S, K_S)$	0.0792	0.1143	0.1294	0.1328	0.1139
		Output of $\mathrm{SDSA}(\cdot)$, $\hat{V}_S$	0.0297	0.0414	0.0456	0.0508	0.0419
	MLP	Layer 1	0.3675	0.4263	0.4505	0.4555	0.4250
		Layer 2	0.0463	0.0532	0.0520	0.0541	0.0514
Block 2	SDSA	Input	0.3493	0.4002	0.4320	0.4391	0.4051
		V_S	0.2582	0.2761	0.2476	0.2237	0.2514
		Q_S	0.0147	0.0191	0.0195	0.0190	0.0181
		K_S	0.0128	0.0172	0.0186	0.0199	0.0171
		$g(Q_S, K_S)$	0.1033	0.1347	0.1357	0.1202	0.1235
		Output of $\mathrm{SDSA}(\cdot)$, $\hat{V}_S$	0.03318	0.04373	0.03913	0.0324	0.0371
	MLP	Layer 1	0.3484	0.3944	0.4259	0.4340	0.4007
		Layer 2	0.0317	0.0404	0.0417	0.0433	0.0393
Block 3	SDSA	Input	0.3454	0.3890	0.4240	0.4292	0.3969
		V_S	0.3018	0.3055	0.2614	0.2193	0.2720
		Q_S	0.0108	0.0151	0.0158	0.0160	0.0144
		K_S	0.0113	0.0152	0.0151	0.0144	0.0140
		$g(Q_S, K_S)$	0.1273	0.1600	0.1569	0.1375	0.1454
		Output of $\mathrm{SDSA}(\cdot)$, $\hat{V}_S$	0.0446	0.0562	0.0462	0.0344	0.0453
	MLP	Layer 1	0.3436	0.3825	0.4147	0.4203	0.3903
		Layer 2	0.0261	0.0334	0.0347	0.0352	0.0323
Block 4	SDSA	Input	0.3458	0.3855	0.4191	0.4283	0.3947
		V_S	0.2112	0.2241	0.1941	0.1728	0.2005
		Q_S	0.0062	0.0101	0.0113	0.0117	0.0099
		K_S	0.0061	0.0095	0.0107	0.0120	0.0096
		$g(Q_S, K_S)$	0.0762	0.0979	0.0981	0.0967	0.0922
		Output of $\mathrm{SDSA}(\cdot)$, $\hat{V}_S$	0.0214	0.0289	0.0245	0.0220	0.0242
	MLP	Layer 1	0.3460	0.3837	0.4146	0.4228	0.3918
		Layer 2	0.0208	0.0258	0.0261	0.0259	0.0247

Continued on next page

Table S1 – continued from previous page

			$T=1$	$T=2$	$T=3$	$T=4$	Average
Block 5	SDSA	Input	0.3491	0.3908	0.4228	0.4306	0.3984
		V_S	0.1493	0.1654	0.1491	0.1395	0.1508
		Q_S	0.0048	0.0080	0.0090	0.0093	0.0078
		K_S	0.0042	0.0071	0.0081	0.0082	0.0069
		$g(Q_S, K_S)$	0.0473	0.0698	0.0740	0.0749	0.0665
		Output of $\mathrm{SDSA}(\cdot)$, $\hat{V}_S$	0.0102	0.0169	0.0157	0.0147	0.0144
	MLP	Layer 1	0.3541	0.3935	0.4231	0.4302	0.4002
		Layer 2	0.0169	0.0205	0.0205	0.0206	0.0196
Block 6	SDSA	Input	0.3614	0.3957	0.4201	0.4258	0.4007
		V_S	0.0729	0.0791	0.0767	0.0778	0.0766
		Q_S	0.0012	0.0021	0.0027	0.0032	0.0023
		K_S	0.0008	0.0018	0.0024	0.0026	0.0019
		$g(Q_S, K_S)$	0.0128	0.0227	0.0260	0.0286	0.0225
		Output of $\mathrm{SDSA}(\cdot)$, $\hat{V}_S$	0.0018	0.0040	0.0043	0.0045	0.0036
	MLP	Layer 1	0.3690	0.4027	0.4264	0.4317	0.4074
		Layer 2	0.0147	0.0180	0.0183	0.0186	0.0174
Block 7	SDSA	Input	0.3619	0.4069	0.4192	0.4218	0.4025
		V_S	0.0379	0.0359	0.0371	0.0406	0.0379
		Q_S	0.0001	0.0002	0.0003	0.0004	0.0003
		K_S	0.0001	0.0003	0.0004	0.0005	0.0003
		$g(Q_S, K_S)$	0.0022	0.0046	0.0058	0.0073	0.0050
		Output of $\mathrm{SDSA}(\cdot)$, $\hat{V}_S$	0.0001	0.0005	0.0005	0.0006	0.0004
	MLP	Layer 1	0.3575	0.4035	0.4156	0.4180	0.3987
		Layer 2	0.0140	0.0184	0.0186	0.0189	0.0175
Block 8	SDSA	Input	0.2865	0.3888	0.4019	0.4106	0.3720
		V_S	0.0200	0.0342	0.0380	0.0419	0.0335
		Q_S	0.00001	0.0001	0.0001	0.0002	0.0001
		K_S	$1e^{-5}$	$1e^{-5}$	0.0001	0.0001	$1e^{-5}$
		$g(Q_S, K_S)$	$2e^{-5}$	0.0001	0.0020	0.0024	0.0015
		Output of $\mathrm{SDSA}(\cdot)$, $\hat{V}_S$	$1e^{-5}$	0.0002	0.0002	0.0002	0.0001
	MLP	Layer 1	0.2716	0.3721	0.3827	0.3899	0.3541
		Layer 2	0.0056	0.0111	0.0110	0.0115	0.0098
Head	FC		0.0002	0.3876	0.3604	0.4843	0.3081

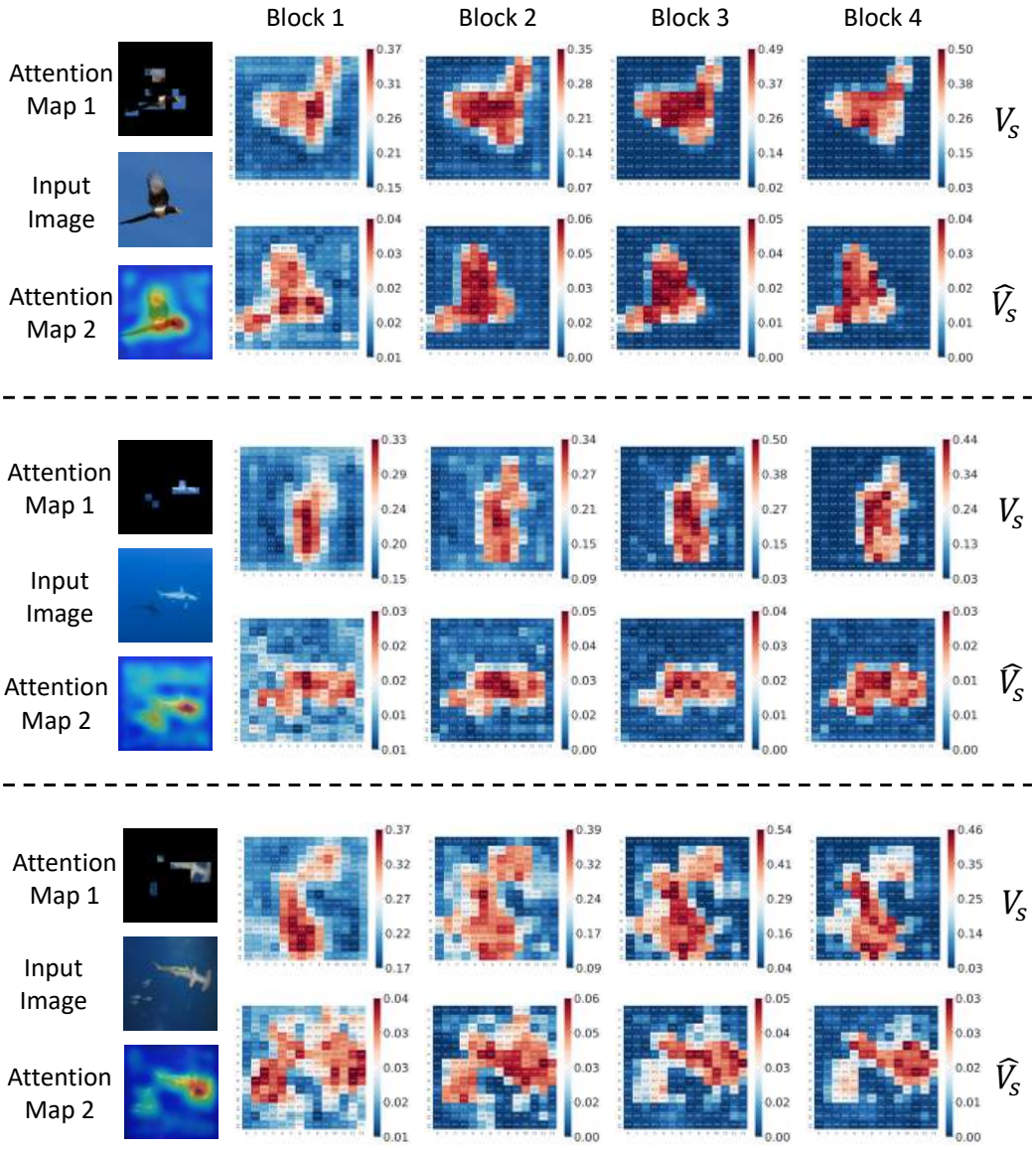

Figure S1: Attention Map Based on Spike Firing Rate (SFR). Attention map 1 and 2 are generated by the Grad-CAM method [95]. V_S is the Value tensor. $\hat{V}_S$ is the output of $\mathrm{SDSA}(\cdot)$. The spike-driven self-attention mechanism masks unimportant channels in V_S to obtain $\hat{V}_S$. Each pixel on V_S and $\hat{V}_S$ represents the SFR at a patch. The spatial resolution of each attention map is 14×14 (196 patches). The redder the higher the SFR, the bluer the smaller the SFR. We can see that the $\mathrm{SDSA}(\cdot)$ regulation of spike firing is basically consistent with the focused points in the attention map.