Spiking Transformer with Spatial-Temporal Attention

Supplementary Material

A. Energy Calculation Details

To clarify the energy consumption of our STAtten architecture in Section 5.3, we present the detailed equations of every layer shown in Table 7.

Table 7. The detailed equations of energy consumption on every layer of STAtten architecture.

Block	Layer	Energy Consumption
Embedding	First Conv	$E_{MAC} \cdot F_{Conv} \cdot T$
	Other Convs	$E_{AC} \cdot F_{Conv} \cdot T \cdot S_{Conv}$
Attention	Q, K, V	$3 \cdot E_{AC} \cdot F_{Conv} \cdot T \cdot S_{Conv}$
	Self-attention	$E_{AC} \cdot TND^2 \cdot (S_K + S_V + S_Q)$
	MLP	$E_{AC} \cdot F_{Conv} \cdot T \cdot S_{Conv}$
MLP	MLP1	$E_{AC} \cdot F_{Conv} \cdot T \cdot S_{Conv}$
	MLP2	$E_{AC} \cdot F_{Conv} \cdot T \cdot S_{Conv}$

Here, E_{MAC} is the energy of MAC operation, F_{Conv} is FLOPs of the convolutional layer, T, N, and D are timestep, the number of patches, and channel dimension respectively, S_{Conv} is the firing rate of input spikes on the convolutional layer, S_Q, S_K, and S_V are the firing rate of input spikes on Q, K, and V projection layer respectively. The FLOPs of the convolutional layer can be calculated as follows:

$$F_{Conv} = K \cdot K \cdot H_{out} \cdot W_{out} \cdot C_{in} \cdot C_{out}, \tag{14}$$

where K is kernel size, H_{out} and W_{out} are the height and width of the output feature map respectively, and C_{in} and C_{out} are the input and output channel dimension respectively.

In the embedding block, for the first convolutional layer, since we use direct coding to convert a float pixel value into binary spikes [49], the firing rate does not need to be calculated for energy consumption, and E_{MAC} is used for the float pixel input. In the Attention block, for the energy calculation of the self-attention part, we can use the equations of our spatial-temporal methods shown in Table 1. Following previous works [53, 64], we calculate the energy consumption based on the FLOPs operation executed in 45nm CMOS technology [18], *e.g.*, $E_{MAC} = 4.6pJ$, and $E_{AC} = 0.9pJ$. The firing rate and the theoretical energy consumption of the STAtten with Spike-driven Transformer architecture are provided in Appendix E.

B. Experimental Details

In this section, we provide the experimental details on CIFAR10/100, ImageNet, CIFAR10-DVS, and N-Caltech101 datasets. The Table 8 shows that general experimental setup in [53]. In other architecture [42, 52, 62, 64], we follow their configurations for fair comparison.

Table 8. The experimental details on each dataset. L-D in architecture represents L number of encoder blocks and D channel dimension.

	CIFAR10/100	ImageNet	DVS
Timestep	4	4	16
Batch size	64	32	16
Learning rate	0.0003	0.001	0.01
Training epoch	310	210	210
Optimizer	AdamW	Lamb	AdamW
Hardware (GPU)	A5000	A100	A5000
Architecture	2-512	8-768	2-256

We apply data augmentation following [53, 64]. For the ImageNet dataset, general augmentation techniques such as random augmentation, mixup, and cutmix are employed. Different data augmentation strategies are applied to the CIFAR10-DVS and N-Caltech101 datasets according to NDA [32]. While training on the dynamic datasets, we add a pooling layer branch and a residual connection to the spatial-temporal attention layer. The outputs of the pooling layer and the spatial-temporal attention are then multiplied element-wise to extract important spike feature maps.

C. Ablation Study

In this section, we analyze the impact of timestep combinations and block sizes in our block-wise attention mechanism.

C.1. Timestep Combination

In section 4.1, we identified that binary matrix multiplication between temporally distant spikes can increase silent neurons, leading to information loss. This phenomenon can be explained through binary matrix multiplication patterns. Let $\mathbf{Q}_t, \mathbf{K}_{t'} \in \{0, 1\}^{N \times D}$ be binary spike matrices at timesteps t and t'. When computing attention between these timesteps, each element of their product is:

$$(\mathbf{Q}_t \mathbf{K}_{t'}^\top)_{i,j} = \sum_{d=1}^{D} q_{t,i,d} \cdot k_{t',j,d}, \tag{15}$$

where $i, j \in \{1, ..., N\}$ represent token positions, and $d \in \{1, ..., D\}$ is the feature dimension. As the temporal distance $|t - t'|$ increases, the spike patterns become less correlated, increasing the probability of $q_{t,i,d} \cdot k_{t',j,d} = 0$. This multiplicative effect accumulates across the dimension D,

Table 9. Analysis of temporal block combinations and their accuracy. Each entry shows timestep ranges for Q/K/V tensors across two blocks (B_1, B_2). For example, [1,2]/[3,4]/[1,2] indicates Q and V use timesteps 1-2 while K uses timesteps 3-4. Notation [0:16] represents timesteps from 0 through 16.

Dataset	Temporal Combination ($\mathbf{Q}/\mathbf{K}/\mathbf{V}$)		Accuracy (%)
	B_1	B_2	
CIFAR100	[1,2] / [1,2] / [1,2]	[3,4] / [3,4] / [3,4]	79.85
	[1,2] / [3,4] / [1,2]	[3,4] / [1,2] / [3,4]	79.28
	[1,4] / [2,3] / [1,4]	[2,3] / [1,4] / [2,3]	79.09
Sequential CIFAR100	[0:16] / [0:16] / [0:16]	[16:32] / [16:32] / [16:32]	62.95
	[0:16] / [16:32] / [0:16]	[16:32] / [0:16] / [16:32]	62.80
N-Caltech101	[0:8] / [0:8] / [0:8]	[8:16] / [8:16] / [8:16]	82.49
	[0:8] / [8:16] / [0:8]	[8:16] / [0:8] / [8:16]	79.09

leading to more zero outputs and consequently more silent neurons.

To illustrate this effect, consider binary matrices $\mathbf{Q}_t$ and $\mathbf{K}_{t'}$ with the same number of spikes but at different temporal distances. For nearby timesteps t and $t+1$:

$$\mathbf{Q}_t = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}, \mathbf{K}_{t+1} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix} \tag{16}$$

Their product yields many high values due to similar patterns:

$$\mathbf{Q}_t\mathbf{K}_{t+1}^\top = \begin{bmatrix} 3 & 2 & 2 & 3 \\ 2 & 3 & 2 & 1 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{bmatrix} \tag{17}$$

However, for distant timesteps t and $t+\Delta$:

$$\mathbf{Q}_t = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}, \mathbf{K}_{t+\Delta} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{18}$$

Their product contains low values and zeros despite having the same spike density:

$$\mathbf{Q}_t\mathbf{K}_{t+\Delta}^\top = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \tag{19}$$

Since we apply LIF after $\mathbf{QK}^\top\mathbf{V}$ operations to generate spikes, matrices with higher values from nearby timesteps are more likely to trigger neurons compared to lower values from distant timesteps. This example demonstrates that temporal distance leads to less correlated spike patterns, resulting in increased silent neurons. Fig. 3(b) visualizes this effect on the CIFAR100 dataset, showing higher neuron activation when correlating nearby timesteps compared to distant ones.

Table 10. Accuracy comparison with different block sizes. T represents the timestep for each dataset, B denotes the block size.

Dataset	Block size	Accuracy (%)
CIFAR100 ($T=4$)	B=2	79.85
	B=4	79.90
ImageNet ($T=4$)	B=1	77.65
	B=2	78.00
	B=4	78.06
Sequential CIFAR100 ($T=32$)	B=8	60.89
	B=16	62.95
	B=32	64.30
N-Caltech101 ($T=16$)	B=4	83.15
	B=8	82.49
	B=16	82.40

Table 9 shows the performance comparison across different datasets by varying temporal combinations of $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$. The notation [a,b]/[c,d]/[e,f] indicates that $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ use timesteps [a,b], [c,d], and [e,f], respectively. For instance, in CIFAR100's B_1, [1,2]/[3,4]/[1,2] means $\mathbf{Q}$ and $\mathbf{V}$ use timesteps 1-2 while $\mathbf{K}$ uses timesteps 3-4. Across all datasets, combinations using different timestep ranges consistently show lower performance compared to those using the same ranges.

C.2. Block Size Analysis

STAtten employs block-wise processing for memory efficiency. Table 10 shows how block size affects performance across different datasets. For CIFAR100 (T=4), using block size B=2 achieves comparable accuracy to full spatial-temporal correlation (B=4), with only 0.05% difference. Similarly, for ImageNet (T=4), block sizes B=2, and B=4 yield accuracies of 78.00%, and 78.06%, indicating that larger block sizes slightly improve performance. However, sequential CIFAR100 (T=32) shows an opposite trend: smaller block sizes lead to decreased accuracy because temporal information dominates spatial features in this dataset. Therefore, we use B=32 for the results presented in Table 2. For N-Caltech101 (T=16), we achieve optimal performance with B=4. This reveals that optimal block size depends on temporal-to-spatial information ratio: vision tasks favor smaller blocks to preserve spike correlation, while sequential tasks need larger blocks for temporal modeling.

D. Versatility in Vision Tasks

To demonstrate the generalizability and robustness of our STAtten, we extend its application to additional vision tasks, including object detection and transfer learning.

D.1. Object Detection

We evaluate the adaptability of STAtten in the object detection domain by integrating it as the backbone in the EMS-YOLO [44] framework, replacing the original backbone. We train the model on the PASCAL VOC dataset [12] from scratch, maintaining the same training configuration as the baseline [44] for a fair comparison. The results, presented in Table 11, demonstrate STAtten's competitive performance in object detection compared to other spike-based architectures [21, 53]. These results highlight STAtten's adaptability to diverse vision tasks beyond classification.

D.2. Transfer Learning

To further validate the generalizability of STAtten, we conduct transfer learning experiments on CIFAR-10 and CIFAR-100 datasets. We leverage pre-trained weights from ImageNet and resize the input images to 224×224 pixels to align with standard transfer learning protocols. The results, also shown in Table 12, indicate that STAtten achieves top performance in transfer learning tasks. These results underscore STAtten's ability to generalize effectively across datasets and tasks, leveraging its spatial-temporal attention mechanism to extract robust features from pre-trained weights.

Table 11. Performance comparison between STAtten and previous works on object detection using PASCAL VOC dataset.

Method	mAP@0.5 (%)	mAP@0.5:0.9 (%)
Spiking-YOLO [21]	51.83	-
SDT [53]	51.63	25.31
STAtten + [53]	**52.98**	**27.53**

Table 12. Performance comparison between STAtten and previous works on transfer learning using ImageNet pre-trained weights on CIFAR-10 and CIFAR-100.

Method	CIFAR-10 (%)	CIFAR-100 (%)
Spikformer [64]	97.03	83.83
SpikingReformer [42]	97.40	85.98
STAtten + [53]	**97.76**	**86.67**

E. Firing rate

In this section, we present the firing rate and energy consumption of each layer in Spike-driven Transformer 8-768 architecture with STAtten, pre-trained with the ImageNet dataset. Note that the firing rates represent the firing rate of input spikes for each layer. Additionally, for the firing rate of Self-attention in the table below, we calculate it using the equation: $S_K + S_V + S_Q$.

Block	Layer	$T = 1$	$T = 2$	$T = 3$	$T = 4$	Energy (mJ)
Embedding	1st Conv	-	-	-	-	0.5982
	2nd Conv	0.0771	0.1389	0.1092	0.1561	0.9015
	3rd Conv	0.0424	0.0644	0.0586	0.0527	0.4089
	4th Conv	0.0328	0.0501	0.0428	0.0480	0.3253
	5th Conv	0.0660	0.1402	0.1308	0.1413	0.4478
Encoder-1	Q, K, V	0.2159	0.2662	0.2609	0.2728	0.3171
	Self-attention	0.1221	0.1313	0.1320	0.1451	0.0993
	MLP	0.2018	0.2962	0.2880	0.3454	0.1177
MLP-1	MLP1	0.3292	0.3605	0.3622	0.3697	0.5916
	MLP2	0.0340	0.0409	0.0401	0.0458	0.0670
Encoder-2	Q, K, V	0.3268	0.3583	0.3543	0.3967	0.4482
	Self-attention	0.0986	0.0950	0.0945	0.1017	0.0867
	MLP	0.2760	0.3532	0.3371	0.3511	0.1370
MLP-2	MLP1	0.3094	0.3332	0.3321	0.3718	0.5604
	MLP2	0.0226	0.0293	0.0301	0.0350	0.0487
Encoder-3	Q, K, V	0.3240	0.3462	0.3504	0.3917	0.4408
	Self-attention	0.0752	0.0694	0.0680	0.0654	0.0772
	MLP	0.2837	0.3409	0.3254	0.2879	0.1288
MLP-3	MLP1	0.3486	0.3519	0.3588	0.3957	0.6056
	MLP2	0.0186	0.0241	0.0245	0.0255	0.0386
Encoder-4	Q, K, V	0.3532	0.3570	0.3661	0.4015	0.4613
	Self-attention	0.0716	0.0707	0.0704	0.0743	0.0749
	MLP	0.2586	0.3299	0.3246	0.3203	0.1283
MLP-4	MLP1	0.3591	0.3544	0.3633	0.3965	0.6132
	MLP2	0.0138	0.0177	0.0183	0.0188	0.0286
Encoder-5	Q, K, V	0.3599	0.3588	0.3688	0.3979	0.4637
	Self-attention	0.0701	0.0619	0.0610	0.0631	0.0694
	MLP	0.2695	0.2588	0.2469	0.2187	0.1034
MLP-5	MLP1	0.3645	0.3579	0.3691	0.3979	0.6199
	MLP2	0.0098	0.0126	0.0128	0.0134	0.0202
Encoder-6	Q, K, V	0.3737	0.3621	0.3706	0.3941	0.4684
	Self-attention	0.0740	0.0581	0.0533	0.0496	0.0606
	MLP	0.2071	0.2260	0.1896	0.1393	0.0793
MLP-6	MLP1	0.3832	0.3665	0.3743	0.3963	0.6327
	MLP2	0.0108	0.0128	0.0119	0.0107	0.0193
Encoder-7	Q, K, V	0.3815	0.3665	0.3663	0.3826	0.4672
	Self-attention	0.0746	0.0575	0.0538	0.0528	0.0615
	MLP	0.1802	0.1670	0.1362	0.0972	0.0604
MLP-7	MLP1	0.3773	0.3574	0.3549	0.3686	0.6069
	MLP2	0.0056	0.0068	0.0068	0.0063	0.0106
Encoder-8	Q, K, V	0.3772	0.3423	0.3471	0.3594	0.4452
	Self-attention	0.1180	0.0853	0.0784	0.0728	0.0926
	MLP	0.1383	0.1324	0.1143	0.1010	0.0505
MLP-8	MLP1	0.3684	0.3480	0.3616	0.3818	0.6075
	MLP2	0.0123	0.0177	0.0168	0.0145	0.0255