SAFformer: Improving Spiking Transformer via Active Predictive Filtering

Zequan Xie[1], **Weiming Zeng**[2], **Yunhua Chen**[1*], **Sichang Lin**[1], **Tongyang Chen**[1] and **Jinsheng Xiao**[3]

[1] School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China
[2]Faculty of Science, Hong Kong Baptist University, China
[3]School of Electronic Information, Wuhan University, China
2112405284@mail2.gdut.edu.cn, 23267062@life.hkbu.edu.hk, yhchen@gdut.edu.cn, fatc811@163.com, chentongyang@mails.gdut.edu.cn, xiaojs@whu.edu.cn

Abstract

Spiking Neural Networks (SNNs) offer notable advantages in biological plausibility and energy efficiency, making them promising candidates for building low-power Transformers. However, existing Spiking Transformers largely adhere to a *passive reactive* paradigm, which struggles to focus on task-relevant information and incurs substantial computational overhead when processing redundant visual data. To overcome this fundamental yet underexplored limitation, we propose SAFformer, a novel Spiking Transformer architecture based on an *active predictive filtering* paradigm. Inspired by the brain's predictive coding mechanism, SAFformer actively suppresses predictable signals and focuses on salient visual features. Extensive experiments show that SAFformer establishes new state-of-the-art performance on CIFAR-10/100 and CIFAR10-DVS. Remarkably, on ImageNet-1K, it achieves 80.44% Top-1 accuracy with only 26.58M parameters and an energy consumption of 5.88 mJ, demonstrating an exceptional balance between accuracy and efficiency.

1 Introduction

Spiking neural networks (SNNs) are regarded as the third generation neural network models[Maass, 1997]. Their event-driven and sparse computing characteristics have demonstrated great potential in the field of artificial intelligence, especially in the small-scale edge computing scenarios[Guo *et al.*, 2022]. Unlike artificial neural networks (ANNs) that perform intensive computations at each inference step, SNNs transmit information asynchronously, with spiking neurons consuming energy only when receiving or emitting spikes[Eshraghian *et al.*, 2023]. In recent years, the integration of SNNs with the powerful Transformer architecture has shown great promise, paving the way for a new generation of visual models that achieve high performance with improved energy efficiency.

Despite this promise, a fundamental limitation persists in the design of existing Spiking Transformers. Whether in early direct-conversion attention mechanisms or later linear-complexity variants like QK-Attention[Zhang *et al.*, 2024], the information selection process universally follows a "Passive Reactive" paradigm. This model involves indiscriminate and costly processing of all information, followed by a post-hoc filtering step that relies solely on the fixed threshold of spiking neurons[Guo *et al.*, 2025; Chen *et al.*, 2024]. Crucially, it lacks the ability to dynamically adjust its computational scope based on the input global context. Consequently, when faced with simple images containing large amounts of redundant background, the model still wastes significant computational resources on irrelevant regions. This paradigm runs counter to the highly efficient information processing mechanisms of the biological brain.

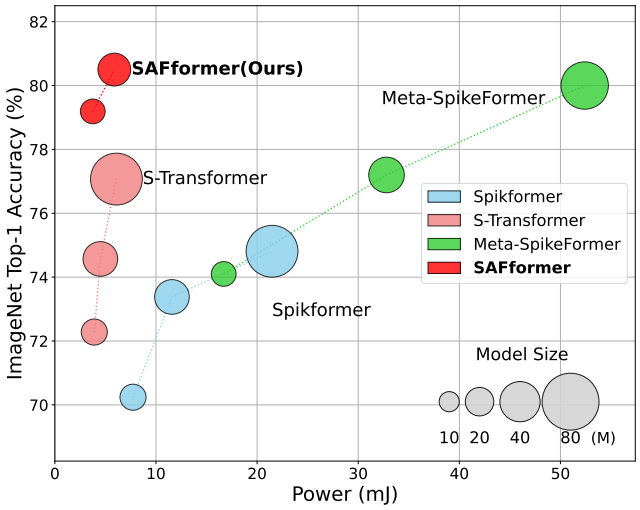

Figure 1: Comparison of our SAFformer with other Spiking Transformers on ImageNet-1k with an input resolution of 224 × 224.

In contrast, the predictive filtering mechanism inherent in the visual system enables the brain to utilize top-down feedback signals to modulate the activity of lower-level neurons[Crassidis and Markley, 1997; Rao and Ballard, 1999]. It actively suppresses predictable, redundant information and only forwards significant prediction errors for subsequent processing, thereby achieving extreme energy efficiency. Specifically, higher-level cortical areas (e.g. V4, IT)

*Corresponding authors.

form a holistic prediction of the scene based on prior experience and send these predictive signals down to lower-level areas (e.g. V1, V2)[Friston, 2009]. The lower-level cortex then compares this prediction with the actual visual input, and only the significant discrepancies are propagated upward as feedback signals, as illustrated in Fig. 3(b).

Beyond the self-attention module, the MLP layer in the Transformer architecture, as an important part of network feature enhancement, is also a centralized area for computation and parameters. Vanilla Spiking MLPs, typically direct conversions of their ANN counterparts, suffer from two major flaws. First, they are spatially blind. Their 1×1 convolutional structure can only mix information along the channel dimension[Wang *et al.*, 2024] and is entirely incapable of perceiving or utilizing the local spatial context of tokens. Second, their feature transformation is static and homogeneous, applying an identical, unconditional transformation to all spatial locations[Chen *et al.*, 2026b]. This inefficient processing model contradicts the principle of the visual cortex, which adjusts its processing intensity based on predictive information.

To systematically address these challenges, we draw inspiration from the brain's active prediction mechanism to propose SAFformer, a novel Spiking Transformer architecture that operates on the principle of Active Predictive Filtering. By incorporating two synergistic, bio-inspired innovations, SAFformer shifts the paradigm from passive reaction to active prediction. Our contributions are as follows:

1) We propose the Spiking Active Predictive Filtering Attention(SAF), a new sparse spiking self-attention mechanism, as shown in Fig. 2. It introduces a lightweight Spiking Sparsity Guidance Module (SGM) to actively predict the saliency distribution of the input, thereby prospectively focusing attention resources on key spatial locations and realizing active predictive filtering.

2) We propose the Spiking Multi-scale Adaptive Gated (SMAG) network to replace the vanilla SMLP. SMAG perceives local spatial context through a decoupled, multi-scale gating mechanism, achieving more refined extraction and selection of predictive features while significantly reducing the number of parameters.

3) Comprehensive experiments demonstrate that our architecture outperforms or matches SOTA Spiking Transformers on various datasets, with an excellent balance of efficiency and performance, as shown in Fig. 1. Notably, we achieve SOTA accuracies of 97.50% on CIFAR-10 and 83.38% on CIFAR-100, setting a new benchmark for the SNN field.[1]

2 Related Work

2.1 Spiking Neural Networks

Spiking Neural Networks (SNNs) are widely recognized as the third generation of neural network models, with their design inspired by the information processing mechanisms of the biological brain. Unlike traditional ANNs, SNNs communicate by transmitting binary spikes, mimicking the firing of biological neurons[Duan *et al.*, 2023]. When the membrane potential of a spiking neuron exceeds a certain threshold, it fires a spike, represented as '1'; otherwise, it remains silent, represented as '0'[Yao *et al.*, 2024b]. This event-driven mechanism facilitates sparse synaptic operations and avoids costly multiply-accumulate (MAC) operations[Chen *et al.*, 2026a], thereby significantly enhancing the energy efficiency of these models. SNNs are primarily trained using two methods: ANN-to-SNN conversion and direct training with surrogate gradients[Di Yu *et al.*, 2024]. ANN-to-SNN conversion involves training an ANN and then converting it to a corresponding quantized SNN version. Direct training, on the other hand, employs a surrogate function to approximate the spike firing process during backpropagation[Yao *et al.*, 2024b]. The effectiveness of direct training in handling temporal data has made it a preferred choice in SNN research.

However, conventional direct training methods require substantial GPU memory due to the simulation of multiple timesteps[Luo *et al.*, 2024; Qiu *et al.*, 2025]. To address the challenge of long training times, Yao introduced a novel direct training method called Spike Firing Approximation (SFA)[Yao *et al.*, 2025]. This method uses quantized membrane potentials as activation values during the training phase and converts these values into spike sequences during inference, thereby reducing training duration. It has been demonstrated that SFA can reduce training time by at least $4.1\times$.

2.2 Spiking Vision Transformers

The Transformer, with its powerful sequence modeling capabilities and self-attention mechanism, has become a mainstream backbone in the field of computer vision[Dosovitskiy *et al.*, 2020]. However, its quadratic computational complexity and significant memory footprint limit its application in power-sensitive and resource-constrained scenarios[Xiao *et al.*, 2026]. Combining the powerful capabilities of the Transformer with the low-power and event-driven characteristics of SNNs has become an active research direction.

Spikformer[Zhou *et al.*, 2023] pioneered spiking self-attention computation, establishing the first spiking ViT. Spikingformer[Zhou *et al.*, 2026] further enhanced the SNN compatibility of the model by designing a pre-activation residual connection. To achieve a fully spike-driven architecture, Spike-driven Transformer[Yao *et al.*, 2023] implemented the Hadamard product in its self-attention module, becoming the first directly trained, fully spike-driven transformer model. To reduce temporal complexity, QKFormer[Zhang *et al.*, 2024] proposed using only Q and K for self-attention computation, becoming the first spiking Transformer with a linear-complexity self-attention module. Additionally, OST[Song *et al.*, 2024] compresses the time domain into 1 to enhance the training speed. To fully address the issue of over-allocating attention, SpiLiFormer[Zheng *et al.*, 2025] introduced a lateral inhibition mechanism to simulate the inhibitory-excitation interaction process.

3 Methods

In this section, we first introduce the spiking neuron model we used. Next, we present the SAFformer architecture, which includes a novel spiking active predictive filtering attention mechanism and a multi-scale gated feedforward network.

[1]The full version is available at https://arxiv.org/abs/2605.08270.

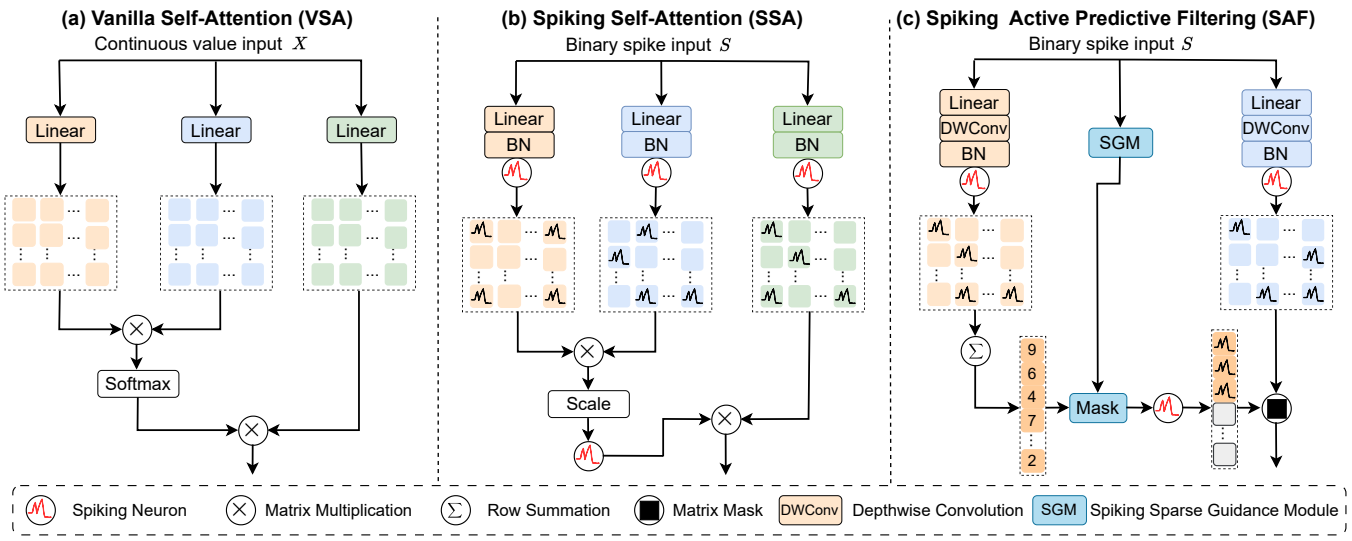

Figure 2: Comparison of three self-attention paradigms. (a)VSA employs floating-point matrix multiplication to evaluate the spatial correlations between Q and K with quadratic computational complexity. (b)SSA transforms the floating-point operations in VSA into spiking computations, maintaining the same complexity as VSA. (c)SAF introduces a predictive filtering paradigm. It maintains linear complexity by using SGM to predict and sparsify significance scores from Q, which then act upon K element-wise.

3.1 Preliminary

In this paper, we use the widely adopted LIF model [Guo *et al.*, 2024], which can be described by the following formula:

$$U[t] = \beta H[t-1] + X[t], \tag{1}$$

$$S[t] = \mathrm{Hea}(U[t] - V_{\mathrm{th}}), \tag{2}$$

$$H[t] = \begin{cases} U[t] \cdot (1 - S[t]) + V_{\mathrm{reset}} \cdot S[t], & \text{hard reset}, \\ U[t] - V_{\mathrm{th}} \cdot S[t], & \text{soft reset}, \end{cases} \tag{3}$$

where t is the time step, $U[t]$ is the membrane potential, the spatial input $X[t]$ is extracted from the original spike sequence via Conv or Linear layers, and the temporal input $\beta H[t-1]$ comes from the decayed membrane potential of the previous time step, β is the leak factor. $S[t]$ is the output spike, and $H[t]$ is the membrane potential after firing. $\mathrm{Hea}(\cdot)$ is the Heaviside step function, where $\mathrm{Hea(x)} = 1$ if $x \geq 0$. When $U[t]$ exceeds the threshold V_{th}, the spiking neuron fires a spike, and the membrane potential is reset to V_{reset}.

3.2 Overall Architecture

The overall architecture of SAFformer is shown in Fig. 3(a), which consists of three stages. The number of blocks (SAFformer blocks or Spikformer blocks) in each stage is represented as N_1, N_2, and N_3, respectively. The input data is represented as $I \in \mathbb{R}^{T \times C_{in} \times H \times W}$, where $C_{in} = 3$ for static image data and $C_{in} = 2$ for neuromorphic data. In the first stage, a patch size of 4×4 is utilized. A Patch Embedding module converts the input into a spike sequence and maps the feature of each patch to an initial channel dimension C, reducing the number of tokens to $\frac{H}{4} \times \frac{W}{4}$.The transformed features are then processed through SAF and SMAG. The processing mechanism in the second stage is similar, but with a patch size of 2×2, an expanded channel dimension of $2C$, and a reduced token count of $\frac{H}{8} \times \frac{W}{8}$. The third stage is the deep stage, where we continue to use a 2×2 patch size, expand the channel dimension to $4C$, and reduce the token count to $\frac{H}{16} \times \frac{W}{16}$.

3.3 Spiking Active Predictive Filtering(SAF)

We propose Spiking Active Predictive Filtering Attention (SAF), whose detailed structure is shown in Fig. 3(c). The core idea of SAF lies in that it no longer calculates the pairwise relationships between tokens. Instead, it employs a lightweight guidance module to predict the most information-dense regions and dynamically focuses computational resources on them. Considering that the intrinsic low-pass filtering characteristics of spiking neurons can attenuate high-frequency signals[Yao *et al.*, 2024a; Fang *et al.*, 2025], we introduce a depthwise separable convolution (DWConv) into the generation paths of the spike matrices Q and K. This is to better preserve local details crucial in the shallow stages of the network. For a given input spike sequence $X \in \mathbb{R}^{N \times C}$, where N is the number of tokens and C is the channel dimension, the generation is formulated as:

$$Q = \mathrm{SN}\left(\mathrm{BN}\left(\mathrm{DWConv}\left(XW_Q\right)\right)\right), \tag{4}$$

$$K = \mathrm{SN}\left(\mathrm{BN}\left(\mathrm{DWConv}\left(XW_K\right)\right)\right), \tag{5}$$

where W_Q and W_K represent the linear projections of the input spike sequence X, and $\mathrm{SN}(\cdot)$ is the spiking activation function. Then, we obtain the initial saliency score for each token by summing over the channel dimension of Q:

$$A_t = \sum_{i=0}^{C} Q_{i,j} \tag{6}$$

To actively filter redundant information and adaptively adjust the computational load based on content complexity, we develop a lightweight and learnable module, named Spiking

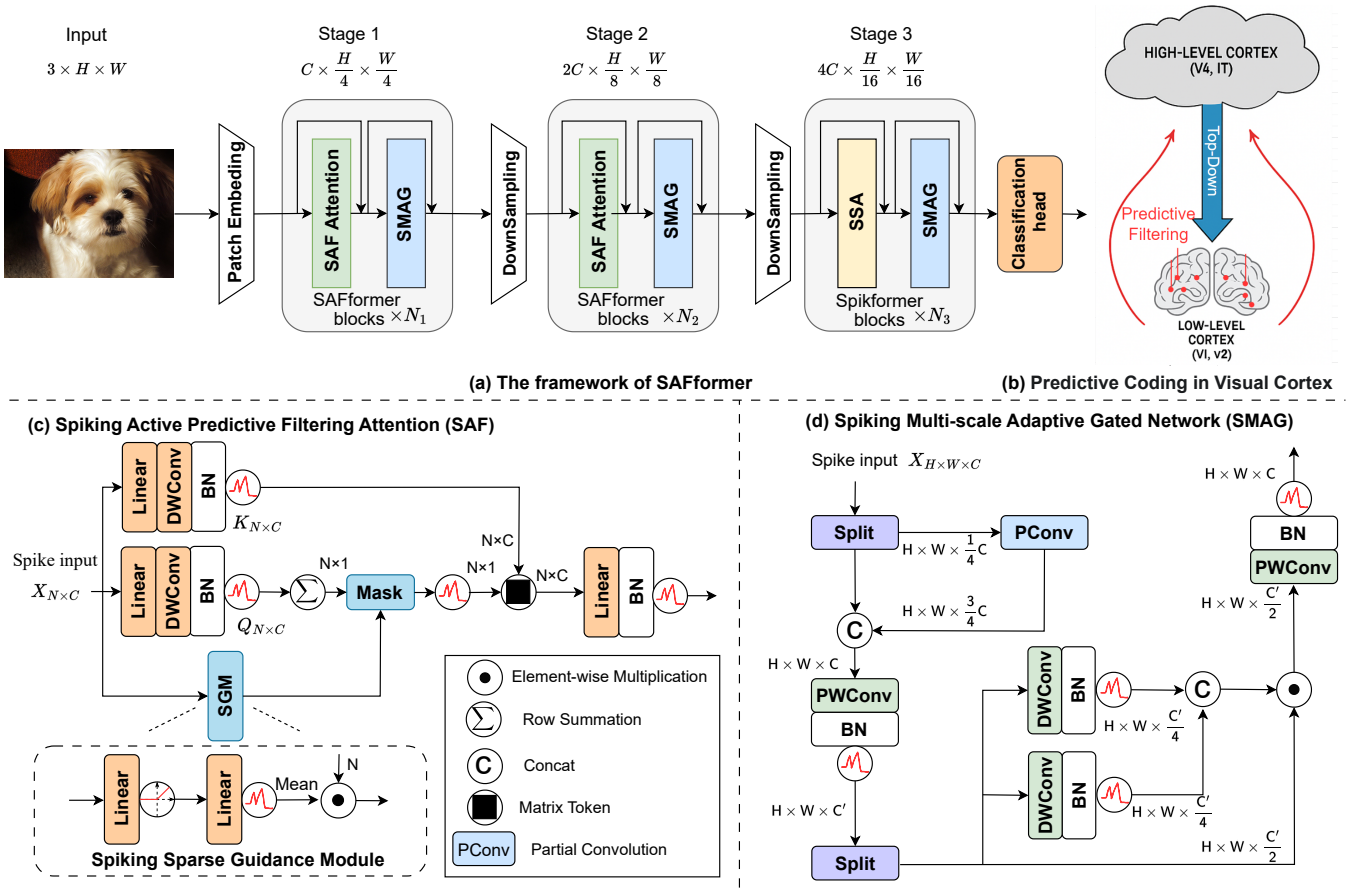

Figure 3: (a) The architecture of SAFformer. We propose the SAFformer block, which consists of SAF Attention and SMAG. (b) An illustration of the predictive coding mechanism in visual cortex. (c) and (d) illustrate the workflows of SAF Attention and SMAG.

Algorithm 1 Spiking Sparsity Guidance Module (SGM)

Input: $X \in \mathbb{R}^{T\times B\times C\times N}$ ▷ Input spike sequence
Output: Dynamic k ▷ The number of tokens to keep
$F \leftarrow \text{Linear}_1(X)$ ▷ Project to hidden space
$F \leftarrow \text{ReLU}(F)$
$G_{raw} \leftarrow \text{Linear}_2(F)$ ▷ Generate guidance scores
$G_{raw} \leftarrow \text{SN}(G_{raw})$ ▷ Shape: $T \times B \times 1 \times N$
$G_{flat} \leftarrow \text{Flatten}(G_{raw})$ ▷ Generate a vector of shape
$p \leftarrow \frac{1}{TBN} \sum_{t,b,n} G_{\text{flat}}[i]$ ▷ Global sparsity ratio
Dynamic $k \leftarrow \lfloor N \cdot p \rfloor$ ▷ Global scaling factor,where N is the token count
return Dynamic k

Sparsity Guidance Module (SGM). It dynamically predict a suitable attention matrix sparsity level, k, based on the global information of the input. In Algorithm 1, we detail the full process of generatingdynamic k using SGM.

With the dynamic k generated by SGM, a masking operation is performed on A_t to create a binary mask matrix M_k:

$$(M_k)_i = \begin{cases} 1, & \text{if } (A_t)_i \in \text{Topk}(A_t, k) \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where Top-K$(\cdot, k)$ denotes the selection of the k largest elements from the vector.

Finally, the mask M_k is applied to the saliency scores, and the result is multiplied element-wise with K in a Hadamard product:

$$\text{SAF}(Q, K) = K \cdot \text{SN}(M_k \odot A_t) \tag{8}$$

Theoretical Analysis

To formally analyze the fundamental differences in information processing across attention mechanisms, we introduce information flow graph[Ahlswede *et al.*, 2000] as a unified analytical framework.

An information flow graph is a weighted directed graph $G = (V, E, W)$, where the vertex set $V = \{v_1, \ldots, v_N\}$ represents the input tokens, a directed edge $(v_j, v_i) \in E$ signifies an information flow from token j to token i, and the weight function $W : E \to \mathbb{R}$ assigns a weight w_{ij} to each edge. The output representation y_i for token i is a weighted combination of information from its inbound neighbors:

$$y_i = f\left(\sum_{v_j \in \text{In}(v_i)} w_{ij} \cdot \text{Info}(v_j)\right) \tag{9}$$

where $\text{In}(v_i)$ is the set of inbound neighbors of v_i, $\text{Info}(v_j)$ is the information carried by token v_j, and $f(\cdot)$ is a non-linear activation function.

Proposition 1. *The computational paradigm of SSA is equivalent to performing a relational aggregation operation on a dense, fully-connected graph.*

Proof. The output y_i of SSA is determined by the equation:

$$y_i = f\left(\sum_{j=1}^{N} \langle Q_i, K_j \rangle \cdot V_j\right) \tag{10}$$

where Q_i, K_j, V_j are the Query,Key,Value vectors derived from tokens i and j. The summation $\sum_{j=1}^{N}$ iterates over all input tokens v_j for any given output y_i. This implies that for any pair of vertices (v_i, v_j), an information flow path, i.e., a directed edge (v_j, v_i), exists. Therefore, the graph is a complete graph where the edge set is $E = V \times V$.

And the weight of an edge (v_j, v_i), denoted w_i, j, is given by the inner product $\langle Q_i, K_j \rangle$. This weight is a binary relational function $R(Q_i, K_j)$ that quantifies the pairwise relation between vertices v_i and v_j. The final output y_i is a weighted sum of the Value information v_j from all vertices in the graph. This process, which gathers information from the entire graph, is a quintessential aggregation operation.

Proposition 2. *The computational paradigm of SAF is equivalent to performing a node-wise filtering operation on a dynamically sparse subgraph.*

Proof. The output y_i of SAF can be represented as:

$$y_i = f\left((M_k)_i \cdot \sum_{c=1}^{C} Q_{c,i} \cdot K_i\right) \tag{11}$$

where q_i, k_i are derived from token i, and $(M_k)_i$ is a dynamic mask. The computation of y_i depends solely on its own derivatives, implying that no cross-vertex edges exist. The information flow graph thus consists only of self-loops.

However, not all vertices participate in the computation. A global selection function $\Psi(X, k)$, realized by the SGM, first identifies an active vertex subset $V_{\text{active}} \subseteq V$.

$$V_{\text{active}} = \{v_i \in V \mid i \in \text{Topk}(\{\sum_{c} Q_{c,j}\}_{j=1}^{N}, k)\} \tag{12}$$

where $|V_{\text{active}}| = k$. The effective computation graph G' is thus a dynamic sparse subgraph consisting only of the vertices in V_{active}.

For any active vertex $v_i \in V_{\text{active}}$, the weight of its self-loop, w_{ii}, is determined by a unary saliency function $S(v_i)$ which is computed as $\sum_{c=1}^{C} Q_{c,i}$. This weight is a function of the vertex's intrinsic properties, not its relation to others. The output y_i is a scaled version of the vertex's own information K_i, which constitutes a filtering operation. Since the selection of the subgraph is dynamically determined by the global input X, the process is an active adaptive filtering.

SAF facilitates a paradigm shift in spiking self-attention, from relational aggregation to dynamic adaptive filtering, offering a new perspective for building efficient and powerful attention mechanisms in Spiking Neural Networks.

3.4 Spiking Multi-scale Adaptive Gated Network(SMAG)

Spiking MLP is a key part of the Spiking Transformer for improving feature representation. However, the commonly used SMLP with wide intermediate channels has the disadvantages of excessive parameter quantity and spatial blind viewing[Qiu *et al.*, 2024], which violates the principle of active predictive filtering that we pursue.

Therefore, we propose a novel feedforward network, the Spiking Multi-scale Adaptive Gated Network (SMAG), to replace the vanilla SMLP, as illustrated in Fig. 3(d). SMAG performs feature transformation in a refine-and-gate paradigm, enhancing performance while reducing the burden of processing redundant information. Specifically, we first introduce a partial convolution[Chen *et al.*, 2023] to a fraction of the channels of the input spike features $X \in \mathbb{R}^{T\times B\times C\times H\times W}$. This operation injects valuable local spatial priors into the feature maps at a minimal computational cost. Subsequently, the features are decoupled into three independent functional paths as follows:

$$X' = \text{SN}\left(\text{BN}\left(W_1 \text{PConv}(X)\right)\right), \tag{13}$$

$$[X'_1, X'_2, X'_3] = X'. \tag{14}$$

where $\text{PConv}(\cdot)$ is the partial convolution operation, and $[,]$ denotes the channel-wise splitting, and W_1 represents a pointwise convolution. PConv effectively introduces local context, which provides a foundation for generating more context-aware gating signals by initially filtering and strengthening a subset of features. Then, X'_1 and X'_2 serve as gating signals, while X'_3 acts as the main path. A multi-scale dynamic gating mechanism is introduced to capture spatial patterns at multiple scales:

$$G'_1 = \text{SN}\left(\text{BN}\left(\text{DWConv}_{3\times3}\left(X'_1\right)\right)\right), \tag{15}$$

$$G'_2 = \text{SN}\left(\text{BN}\left(\text{DWConv}_{7\times7}\left(X'_2\right)\right)\right), \tag{16}$$

$$X'_{\text{gated}} = \text{Concat}(G_1, G_2) \odot X'_3. \tag{17}$$

where $\text{DWConv}_{3\times3}$ and $\text{DWConv}_{7\times7}$ are depthwise convolutions with kernel sizes of 3 and 7, respectively. They learn spatial filters independently for each channel, significantly reducing parameters and computational complexity compared to standard convolutions, while effectively capturing channel-specific spatial patterns. The parallel design of 3x3 and 7x7 kernels enables the gating signals to perceive both fine-grained details and broader regional context simultaneously. Finally, through the element-wise gating multiplication, active suppression and enhancement of the content stream are achieved.

Theoretically, SMAG has significantly fewer parameters than vanilla SMLP. Moreover, compared to the two identical linear transformations in SMLP, the dynamic gating mechanism of SMAG endows the network with the ability to adaptively select and modulate feature channels based on the input content. This allows the model to focus more on critical information and suppress redundancy, thereby improving the discriminability of features.

Methods	Type	Architecture	Param (M)	Time Step	Power (mJ)	Top-1 Acc(%)
DeiT[Touvron *et al.*, 2021]	ANN	DeiT-S	22.00	1	21.20	79.80
	ANN	DeiT-B	86.59	1	80.50	**81.80**
MST[Wang *et al.*, 2023]	A2S	Swin Transformer-T	28.50	512	-	78.51
STA[Hwang *et al.*, 2024]	A2S	ViT-B/32	86.00	32	-	**78.72**
Spikformer[Zhou *et al.*, 2023]	SNN	Spikformer-8-384	16.81	4×1	7.73	70.24
	SNN	Spikformer-8-512	29.68	4×1	11.58	73.38
S-Transformer v2[Yao *et al.*, 2024b]	SNN	S-Transformer v2-8-384	15.10	4×1	16.70	74.10
	SNN	S-Transformer v2-8-512	31.30	4×1	32.80	77.20
Max-Former[Fang *et al.*, 2025]	SNN	Max-10-384	16.23	4×1	4.89	77.82
	SNN	Max-10-512	28.65	4×1	7.49	79.86
MSViT[Hua *et al.*, 2025]	SNN	MSViT-10-384	17.69	4×1	16.65	80.09
	SNN	MSViT-10-512	30.23	4×1	24.74	82.96
E-SpikeFormer[Yao *et al.*, 2025]	SNN	E-SpikeFormer-S	10.00	1×4	3.00	78.50
	SNN	E-SpikeFormer-M	19.00	1×4	5.90	79.80
QSD-Transformer[Qiu *et al.*, 2025]	SNN	S-Transformer v2-T†	1.80	1×4	2.50	77.50
	SNN	S-Transformer v2-M†	3.90	1×4	5.70	78.90
SAFformer	SNN	SAFformer-10-384	15.12	1×4	3.75	79.19
	SNN	SAFformer-10-512	26.58	1×4	5.88	**80.44± 0.24**

Table 1: Evaluation on ImageNet. All models default to an input size of 224 × 224. We have reformatted the time steps of all directly trained SNNs into the $T \times D$ format, where T is the number of time steps, and D is the upper limit of integer activation during training. $D \geq 1$ signifies the use of the SFA direct training method. The † symbol indicates the use of knowledge distillation.

Method	CIFAR10		CIFAR100		CIFAR10-DVS		DVS128	
	Param	Acc	Param	Acc	Param	Acc	Param	Acc
FSTA-SNN[Yu *et al.*, 2025]	11.30	96.52	11.30	80.42	11.20	82.7	-	-
STAA-SNN[Zhang *et al.*, 2025]	-	97.14	-	82.05	-	82.1	-	98.6
Spikformer[Zhou *et al.*, 2023]	9.32	95.51	9.32	78.21	2.57	80.9	2.57	98.3
S-Transformer[Yao *et al.*, 2023]	10.28	95.60	10.28	78.40	2.57	80.0	2.57	**99.3**
QKFormer[Zhang *et al.*, 2024]	6.74	96.18	6.74	81.15	1.50	84.0	1.50	98.6
MSViT[Hua *et al.*, 2025]	7.59	96.53	7.59	81.98	1.67	84.0	1.67	98.8
Max-Former[Fang *et al.*, 2025]	6.57	97.04	6.60	82.65	1.45	84.2	1.45	98.6
SAFformer	6.31	**97.50**	6.34	**83.38**	1.50	**85.0**	1.50	99.0

Table 2: Comparison on CIFAR10, CIFAR100, DVS128, and CIFAR10-DVS.

4 Experiments

In this section, we conduct a comprehensive set of experiments to evaluate our proposed method, comparing it with other recent state-of-the-art architectures on several widely-used benchmarks. We evaluate our model on various datasets, including the static datasets including CIFAR-10/100[Krizhevsky *et al.*, 2009] and ImageNet[Deng *et al.*, 2009], as well as neuromorphic datasets CIFAR10-DVS[Li *et al.*, 2017] and DVS128Gesture[Amir *et al.*, 2017].

4.1 Results on ImageNet-1K Classification

Experimental Setup on ImageNet-1K. The input image size is set to 224 × 224. For model training, we employ the SFA direct training method[Yao *et al.*, 2025] with the virtual timestep D set to 4. We use the AdamW optimizer with a base learning rate of 6×10^{-4}, where the actual learning rate is calculated by multiplying the base rate by the batch size divided by 256. Training is conducted for 200 epochs with a batch size of 512, distributed across 6 NVIDIA RTX 3090 GPUs. Throughout the process, we apply standard data augmentation techniques, including RandAugment[Cubuk *et al.*, 2020], mixup, and random erasing. The number of blocks in the three stages is set to $\{1, 2, 7\}$, respectively.

SAF	SMAG	Param(M)	Top-1 Acc(%)
baseline		6.74	79.79
✓		6.77	80.42(+0.63)
	✓	6.32	83.07(+3.28)
✓	✓	6.34	83.38(+3.59)

Table 3: Ablation study of SAFformer on CIFAR100.

Primary Results on ImageNet-1K. The experimental results are shown in Table 1. Compared with the direct training method of vanilla time steps, our SAFformer (26.58M, 80.44%) achieves a 2.24% higher accuracy than S-Transformer v2 (31.30M, 77.20%) while reducing the number of parameters by approximately 15.1%. Compared with the direct training method of SFA, under the same virtual time

step $D = 4$, our model has a 0.64% higher accuracy rate and similar energy consumption compared with E-SpikeFormer. Compared to the latest Max-Former (28.65M, 79.86%), SAFformer achieves a 0.58% performance improvement with fewer parameters and lower energy consumption. Without considering knowledge distillation, our model achieves comparable accuracy and efficiency to SOTA models in the SNN domain on ImageNet-1K.

4.2 Results on CIFAR and Neuromorphic Datasets

CIFAR Classification. On CIFAR datasets, SAFformer is trained for 400 epochs with a batch size of 64, and the virtual timestep D is set to 4. We use the AdamW optimizer with a learning rate of 1×10^{-3}. The SAFformer consists of 4 blocks, distributed across the three stages as $\{1, 1, 2\}$. The results are shown in Table 2. On CIFAR10, our model achieves 97.50% accuracy with 6.31M parameters, surpassing all previous models including Max-Former (97.04%) and STAA-SNN (97.14%). On the more challenging CIFAR100 dataset, SAFformer achieves an accuracy of 83.38%, which is also significantly higher than the previous SOTA models Max-Former (82.65%) and STAA-SNN (82.05%).

Neuromorphic Classification. The neuromorphic datasets use a SAFformer with 1.50M parameters, configured with a $\{0, 1, 1\}$ block distribution across its three stages and a timestep of 16. The training process for DVS128Gesture involves 200 epochs, while for CIFAR10-DVS it is 106 epochs. On CIFAR10-DVS, our model achieves an accuracy of 85.0%, significantly outperforming QKFormer (84.0%) and Max-Former (84.2%) to establish a new state-of-the-art. On the DVS128 Gesture recognition task, SAFformer achieves an accuracy of 99.0%, demonstrating highly competitive performance.

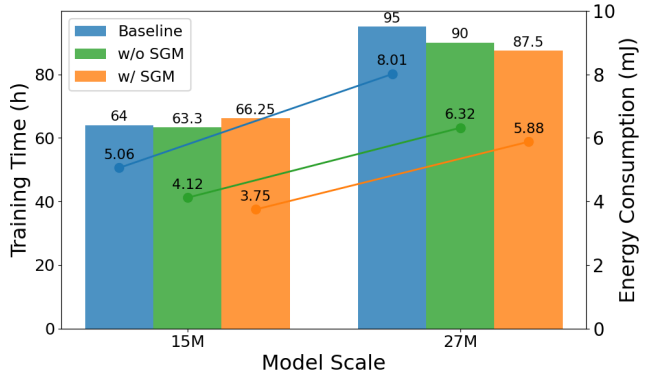

Figure 4: Ablation study on SGM, analyzing its impact on training time and energy consumption across different model scales.

4.3 Ablation Study

We conduct ablation studies on CIFAR100 to analyze the effectiveness of each module, with all experiments following the training details described in Section 4.2 unless otherwise specified. The experimental results are presented in Table 3. The results show that each component contributes positively to the final performance. Specifically, SAF has improved its accuracy by 0.63% with almost no increase in the number of parameters by dynamically predict the tokens with the richest information. Furthermore, substituting the SMLP with our SMAG yields a substantial 3.28% accuracy gain while simultaneously reducing the parameter count by 0.42M. The best performance is achieved when both SAF and SMAG modules are combined, demonstrating that they are effective and complementary.

Methods	Param(M)	Top-1 Acc(%)
w/o depthwise conv	6.77	80.42
w/o partial conv	6.23	82.51
w/o multi-scale	6.36	83.15
SMAG(ours)	**6.34**	**83.38**

Table 4: Ablation study on the components of the SMAG module.

Effectiveness of the SGM Module in SAF. The effectiveness of the SGM module is validated in Fig. 4. We use a SAFformer without any filtering mechanism as the baseline. The "w/o SGM" version removes SGM and uses a fixed Top-K ratio instead. For the 27M model, SGM reduces energy consumption by approximately 27% compared to the baseline and accelerates training time by 8%. Our SAFformer (15.12M, 3.75mJ,79.19%) outperforms the Fixed Top-k baseline (15.08M, 4.12mJ, 78.71%), which demonstrates SGM's effective efficiency and accuracy. While the training time is slightly longer for the smaller 15M model due to relative overhead, the substantial energy savings in all scenarios confirm the superior efficiency of our active predictive approach.

Component Analysis of the SMAG Module. We analyzed the contribution of each key design in SMAG through a step-by-step ablation, with results in Table 4. When we remove the multi-scale design of SMAG, the model's performance drops to 83.15%. Further removing the partial convolution reduces accuracy to 82.51%. Completely eliminating spatial awareness by replacing all depthwise convolutions with 1×1 convolutions causes the most significant performance degradation, with accuracy falling to 80.42%. These results clearly demonstrate that the superiority of SMAG stems from the synergistic combination of its spatial awareness, local prior injection, and multi-scale gating.

5 Conclusion

In this paper, we have delved into the inherent Passive Reactive limitation in existing Spiking Transformers. Drawing inspiration from the brain's predictive coding theory, we proposed SAFformer. It actively predicts and filters redundant information, guiding the model to focus its computational resources on salient features while preserving more high-frequency information. We introduced a new self-attention mechanism, SAF, which actively predicts global saliency via a lightweight guidance module to dynamically focus attention resources on key regions. Furthermore, we proposed SMAG to replace the vanilla SMLP. It leverages a decoupled, multi-scale gating mechanism to overcome homogeneous transformation and parameter redundancy. Experimental results demonstrate that SAFformer achieves state-of-the-art performance on CIFAR-10/100 and CIFAR10-DVS datasets while maintaining low energy consumption.

Acknowledgments

This work was supported by the Natural Science Foundation of Guangdong Province, China (No.2025A1515012243) and the Guangdong S&T programme, China (No.2025A1010010002).

References

[Ahlswede *et al.*, 2000] Rudolf Ahlswede, Ning Cai, S-YR Li, and Raymond W Yeung. Network information flow. *IEEE Transactions on information theory*, 46(4):1204–1216, 2000.

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

[Chen *et al.*, 2023] Jierun Chen, Shiu-hong Kao, Hao He, Weipeng Zhuo, Song Wen, Chul-Ho Lee, and S-H Gary Chan. Run, don't walk: chasing higher flops for faster neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12021–12031, 2023.

[Chen *et al.*, 2024] Yunhua Chen, Ren Feng, Zhimin Xiong, Jinsheng Xiao, and Jian K Liu. High-performance deep spiking neural networks via at-most-two-spike exponential coding. *Neural Networks*, 176:106346, 2024.

[Chen *et al.*, 2026a] Yunhua Chen, Zequan Xie, Jinyu Zhong, Pinghua Chen, and Jinsheng Xiao. Sqkformer: Spiking sparse qkformer with adaptive batch normalization for membrane potential. *Neurocomputing*, page 132666, 2026.

[Chen *et al.*, 2026b] Yunhua Chen, Jinyu Zhong, Yihao Guo, Zequan Xie, Jinsheng Xiao, and Pinghua Chen. C3net: A cross-modal collaborative calibration of features for object detection using frames and events. *Neural Networks*, page 108651, 2026.

[Crassidis and Markley, 1997] John L Crassidis and F Landis Markley. Predictive filtering for nonlinear systems. *Journal of Guidance, Control, and Dynamics*, 20(3):566–572, 1997.

[Cubuk *et al.*, 2020] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[Di Yu *et al.*, 2024] Xin Du Di Yu, Linshan Jiang, Wentao Tong, and Shuiguang Deng. Ec-snn: Splitting deep spiking neural networks for edge devices. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024.

[Dosovitskiy *et al.*, 2020] 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. *arXiv preprint arXiv:2010.11929*, 2020.

[Duan *et al.*, 2023] Peiqi Duan, Yi Ma, Xinyu Zhou, Xinyu Shi, Zihao W Wang, Tiejun Huang, and Boxin Shi. Neurozoom: Denoising and super resolving neuromorphic events and spikes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):15219–15232, 2023.

[Eshraghian *et al.*, 2023] Jason K Eshraghian, Max Ward, Emre O Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D Lu. Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 111(9):1016–1054, 2023.

[Fang *et al.*, 2025] Yuetong Fang, Deming Zhou, Ziqing Wang, Hongwei Ren, ZeCui Zeng, Lusong Li, Shibo Zhou, and Renjing Xu. Spiking transformers need high frequency information. *arXiv preprint arXiv:2505.18608*, 2025.

[Friston, 2009] Karl Friston. The free-energy principle: a rough guide to the brain? *Trends in cognitive sciences*, 13(7):293–301, 2009.

[Guo *et al.*, 2022] Yufei Guo, Liwen Zhang, Yuanpei Chen, Xinyi Tong, Xiaode Liu, YingLei Wang, Xuhui Huang, and Zhe Ma. Real spike: Learning real-valued spikes for spiking neural networks. In *European conference on computer vision*, pages 52–68. Springer, 2022.

[Guo *et al.*, 2024] Yufei Guo, Yuanpei Chen, Xiaode Liu, Weihang Peng, Yuhan Zhang, Xuhui Huang, and Zhe Ma. Ternary spike: Learning ternary spikes for spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 12244–12252, 2024.

[Guo *et al.*, 2025] Yufei Guo, Xiaode Liu, Yuanpei Chen, Weihang Peng, Yuhan Zhang, and Zhe Ma. Spiking transformer: Introducing accurate addition-only spiking self-attention for transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24398–24408, 2025.

[Hua *et al.*, 2025] Wei Hua, Chenlin Zhou, Jibin Wu, Yansong Chua, and Yangyang Shu. Msvit: Improving spiking vision transformer using multi-scale attention fusion. *arXiv preprint arXiv:2505.14719*, 2025.

[Hwang *et al.*, 2024] Sangwoo Hwang, Seunghyun Lee, Dahoon Park, Donghun Lee, and Jaeha Kung. Spikedattention: Training-free and fully spike-driven transformer-to-snn conversion with winner-oriented spike shift for softmax operation. *Advances in Neural Information Processing Systems*, 37:67422–67445, 2024.

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

[Li *et al.*, 2017] 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.

[Luo *et al.*, 2024] Xinhao Luo, Man Yao, Yuhong Chou, Bo Xu, and Guoqi Li. Integer-valued training and spike-driven inference spiking neural network for high-performance and energy-efficient object detection. In *European Conference on Computer Vision*, pages 253–272. Springer, 2024.

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

[Qiu *et al.*, 2024] Xuerui Qiu, Rui-Jie Zhu, Yuhong Chou, Zhaorui Wang, Liang-jian Deng, and Guoqi Li. Gated attention coding for training high-performance and efficient spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 601–610, 2024.

[Qiu *et al.*, 2025] Xuerui Qiu, Malu Zhang, Jieyuan Zhang, Wenjie Wei, Honglin Cao, Junsheng Guo, Rui-Jie Zhu, Yimeng Shan, Yang Yang, and Haizhou Li. Quantized spike-driven transformer. *arXiv preprint arXiv:2501.13492*, 2025.

[Rao and Ballard, 1999] Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87, 1999.

[Song *et al.*, 2024] Xiaotian Song, Andy Song, Rong Xiao, and Yanan Sun. One-step spiking transformer with a linear complexity. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 3142–3150, 2024.

[Touvron *et al.*, 2021] 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.

[Wang *et al.*, 2023] Ziqing Wang, Yuetong Fang, Jiahang Cao, Qiang Zhang, Zhongrui Wang, and Renjing Xu. Masked spiking transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1761–1771, 2023.

[Wang *et al.*, 2024] Yan Wang, Yusen Li, Gang Wang, and Xiaoguang Liu. Multi-scale attention network for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5950–5960, 2024.

[Xiao *et al.*, 2026] Jinsheng Xiao, Hao Ma, Ruidi Chen, Xingyu Gao, Hailong Shi, and Zhongyuan Wang. Stkps-net: Spatio-temporal key patch selection network for few shot anomalous action recognition. *IEEE Transactions on Information Forensics and Security*, 21:827–838, 2026.

[Yao *et al.*, 2023] Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. *Advances in neural information processing systems*, 36:64043–64058, 2023.

[Yao *et al.*, 2024a] Man Yao, Jiakui Hu, Tianxiang Hu, Yifan Xu, Zhaokun Zhou, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer v2: Meta spiking neural network architecture inspiring the design of next-generation neuromorphic chips. *arXiv preprint arXiv:2404.03663*, 2024.

[Yao *et al.*, 2024b] Man Yao, Ole Richter, Guangshe Zhao, Ning Qiao, Yannan Xing, Dingheng Wang, Tianxiang Hu, Wei Fang, Tugba Demirci, Michele De Marchi, et al. Spike-based dynamic computing with asynchronous sensing-computing neuromorphic chip. *Nature Communications*, 15(1):4464, 2024.

[Yao *et al.*, 2025] Man Yao, Xuerui Qiu, Tianxiang Hu, Jiakui Hu, Yuhong Chou, Keyu Tian, Jianxing Liao, Luziwei Leng, Bo Xu, and Guoqi Li. Scaling spike-driven transformer with efficient spike firing approximation training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[Yu *et al.*, 2025] Kairong Yu, Tianqing Zhang, Hongwei Wang, and Qi Xu. Fsta-snn: Frequency-based spatial-temporal attention module for spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22227–22235, 2025.

[Zhang *et al.*, 2024] Han Zhang, Zhaokun Zhou, Liutao Yu, Liwei Huang, Xiaopeng Fan, Li Yuan, Zhengyu Ma, Huihui Zhou, Yonghong Tian, et al. Qkformer: Hierarchical spiking transformer using qk attention. *Advances in Neural Information Processing Systems*, 37:13074–13098, 2024.

[Zhang *et al.*, 2025] Tianqing Zhang, Kairong Yu, Xian Zhong, Hongwei Wang, Qi Xu, and Qiang Zhang. Staa-snn: Spatial-temporal attention aggregator for spiking neural networks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13959–13969, 2025.

[Zheng *et al.*, 2025] Zeqi Zheng, Yanchen Huang, Yingchao Yu, Zizheng Zhu, Junfeng Tang, Zhaofei Yu, and Yaochu Jin. Spiliformer: Enhancing spiking transformers with lateral inhibition. *arXiv preprint arXiv:2503.15986*, 2025.

[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 *ICLR*, 2023.

[Zhou *et al.*, 2026] Chenlin Zhou, Liutao Yu, Zhaokun Zhou, Han Zhang, Jiaqi Wang, Huihui Zhou, Zhengyu Ma, and Yonghong Tian. Spikingformer: A key foundation model for spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 2236–2244, 2026.