TYPE Original Research
PUBLISHED 10 November 2023
DOI 10.3389/fnins.2023.1261543

STCA-SNN: self-attention-based temporal-channel joint attention for spiking neural networks

Xiyan Wu, Yong Song*, Ya Zhou*, Yurong Jiang, Yashuo Bai, Xinyi Li and Xin Yang

School of Optics and Photonics, Beijing Institute of Technology, Beijing, China

OPEN ACCESS

EDITED BY
Lei Deng,
Tsinghua University, China

REVIEWED BY
Malu Zhang,
National University of Singapore, Singapore
Man Yao,
Xi'an Jiaotong University, China
Fangwen Yu,
Tsinghua University, China

*CORRESPONDENCE
Yong Song
✉ yongsong@bit.edu.cn
Ya Zhou
✉ zhouya@bit.edu.cn

RECEIVED 20 July 2023
ACCEPTED 23 October 2023
PUBLISHED 10 November 2023

CITATION
Wu X, Song Y, Zhou Y, Jiang Y, Bai Y, Li X and Yang X (2023) STCA-SNN: self-attention-based temporal-channel joint attention for spiking neural networks.
Front. Neurosci. 17:1261543.
doi: 10.3389/fnins.2023.1261543

Spiking Neural Networks (SNNs) have shown great promise in processing spatio-temporal information compared to Artificial Neural Networks (ANNs). However, there remains a performance gap between SNNs and ANNs, which impedes the practical application of SNNs. With intrinsic event-triggered property and temporal dynamics, SNNs have the potential to effectively extract spatio-temporal features from event streams. To leverage the temporal potential of SNNs, we propose a self-attention-based temporal-channel joint attention SNN (STCA-SNN) with end-to-end training, which infers attention weights along both temporal and channel dimensions concurrently. It models global temporal and channel information correlations with self-attention, enabling the network to learn 'what' and 'when' to attend simultaneously. Our experimental results show that STCA-SNNs achieve better performance on N-MNIST (99.67%), CIFAR10-DVS (81.6%), and N-Caltech 101 (80.88%) compared with the state-of-the-art SNNs. Meanwhile, our ablation study demonstrates that STCA-SNNs improve the accuracy of event stream classification tasks.

KEYWORDS
spiking neural networks, self-attention, temporal-channel, neuromorphic computing, event streams

1. Introduction

As the representatives of mimicking the human brain at the neuronal level, Spiking Neural Networks (SNNs) have gained great attraction for the high biological plausibility, event-driven property, and high energy efficiency (Rieke et al., 1999; Gerstner et al., 2014; Bellec et al., 2018). Using time as an additional input dimension, SNNs record valuable information in a sparse manner and deliver information through spikes only when the membrane potential reaches the firing threshold (Mainen and Sejnowski, 1995). Inspired by biological visual processing mechanisms, Dynamic Vision Sensors (DVS) encode the time, location, and polarity of the brightness changes per pixel into event streams (Lichtsteiner et al., 2008; Posch et al., 2010). With its unique advantages of high event rate, high dynamic range, and fewer resource requirements (Gallego et al., 2020), DVS has broad application prospects in various visual tasks, such as autonomous driving (Cheng et al., 2019), high-speed object tracking (Rebecq et al., 2019), optical flow estimation (Ridwan and Cheng, 2017), and action recognition (Amir et al., 2017). Event-based vision is one of the typical advantage application scenarios of SNNs, providing a platform for demonstrating the capabilities of spiking neurons to process information with spatio-temporal dynamics.

Although the intrinsic time-dependent neuron dynamics endows SNNs with the ability to process spatio-temporal information, there remains a performance gap between SNNs and

ANNs. Recently, ANNs' modules (Hu et al., 2021; Yang et al., 2021; Yao et al., 2021, 2023c) have been integrated into SNNs to improve the performance of SNNs. CSNN (Xu et al., 2018) first validated the application of convolution structure on SNNs, promoting the development of SNNs. Convolution-based SNNs share weights across both temporal and spatial dimensions, following the assumption of spatio-temporal invariance (Huang et al., 2022). This approach can be regarded as a local way of information extraction since convolutional operations can only process a local neighborhood at a time, either in space or time. However, when dealing with sequential data like event streams, capturing long-distance dependencies is of central importance to modeling complex temporal dynamics. Non-local operations (Wang et al., 2018) provided a solution as a building block by computing the response at a position as a weighted sum of the features at all positions. The range of positions can span across space, time, or spacetime, allowing non-local operators to achieve remarkable success in vision attention.

The attention mechanism is inspired by the human ability to selectively find prominent areas in complex scenes (Itti et al., 1998). A popular research direction is to present attention as a lightweight auxiliary unit to improve the representation power of the basic model. In the ANNs domain, Ba et al. (2014) first introduced the term "visual attention" for image classification tasks, utilizing attention to identify relevant regions and locations within the input image. This approach also reduces the computational complexity of the proposed model regarding the size of the input image. SENet (Hu et al., 2018) was introduced to reweight the channel-wise responses of the convolutional features, determining "what" to pay attention to. CBAM (Woo et al., 2018) inferred attention maps sequentially along channel-wise and spatial dimensions for refining the input feature, determining "what" and "where" to pay attention to concurrently. In the SNNs domain, TA-SNN (Yao et al., 2021) first extended the channel-wise attention concept to temporal-wise attention and integrated it into SNNs to determine 'when' to pay attention. MA-SNN (Yao et al., 2023c) extended CBAM to SNNs and proposed a multi-dimensional attention module along temporal-wise, channel-wise, and spatial-wise separately or simultaneously. Recently, TCJA-SNN (Zhu et al., 2022) cooperated temporal-wise and channel-wise attention correlations using the 1-D convolution operation to present the correlation between time-steps and channels. However, the receptive field of TCJA-SNN is a local cross shape that is restricted by its convolution kernels, shown in Figure 1A. Thus long-range dependencies can only be captured when 1-D convolution operation is repeated, which makes multi-hop dependency modeling difficult. On the other hand, self-attention, another vital feature of the human biological system, possesses the ability to capture feature dependencies effectively as an additional non-local operator alongside SE and CBAM. It has sparked a significant wave of interest and achieved remarkable success in various tasks (Vaswani et al., 2017; Dosovitskiy et al., 2020; Liu et al., 2021). Intuitively, there is a compelling interest in investigating the application of self-attention in SNNs to advance deep learning, when considering the biological characteristics of both mechanisms (Yao et al., 2023a,b; Zhou C. et al., 2023; Zhou Z. et al., 2023).

To address the local spatio-temporal receptive field limitation of TCJA, we first adopt self-attention, a non-local operation, to model global temporal and channel information correlations. The self-attention module we employed can capture the global spatio-temporal receptive field, as shown in Figure 1B, allowing for the direct long-range dependencies modeling, which is the highlight of our work. We propose a plug-and-play Self-attention-based Temporal-Channel joint Attention (STCA) module for SNNs with end-to-end training. The STCA-SNNs can learn to focus on different features of the input at each time-step. In other words, the STCA-SNNs can learn 'when' and 'what' to attend concurrently, enhancing the ability of the SNNs to process temporal information. We evaluated the effectiveness of STCA-SNNs across different architectures on three benchmark event stream classification datasets: N-MNIST, CIFAR10-DVS, and N-Caltech 101. Our detailed experiments show that STCA-SNNs achieve competitive accuracy with existing state-of-the-art SNNs.

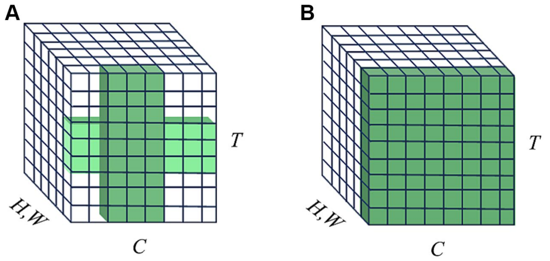

FIGURE 1
Illustration of receptive fields on channel and temporal domains. *T* means the temporal domain, *C* means the channel domain, and *H*, *W* represent the spatial domain. **(A)** TCJA-SNN utilizes two local attention mechanisms with 1-D convolution along temporal-wise and channel-wise, respectively, then fuse them, forming a cross-shaped receptive field. **(B)** STCA-SNN uses self-attention operation to establish temporal-wise and channel-wise correlations, forming a global receptive field.

The main contributions of our work are summarized as follows:

1. We propose STCA-SNNs for event streams that can undertake end-to-end training and inference tasks.
2. The plug-and-play STCA module models global temporal and channel correlations with self-attention, allowing the network to learn 'when' and 'what' to attend simultaneously. This enhances the ability of SNNs to process temporal information.
3. We evaluate the performance of STCA-SNNs on three benchmark event stream classification datasets, N-MNIST, CIFAR10DVS, and N-Caltech 101. Our experimental results demonstrate that STCA-SNNs achieve competitive accuracy compared to existing state-of-the-art SNNs.

2. Related work

2.1. Attention in SNNs

Spiking neural networks benefit from biological plausibility and continuously pursue the combination with brain mechanisms. The attention mechanism draws inspiration from the human ability to selectively identify salient regions within complex scenes and has gained remarkable success in deep learning by allocating attention weights preferentially to the most informative input components. A popular research direction is to present attention as an auxiliary module that can be easily integrated with existing architectures to boost the representation power of the basic model (Hu et al., 2018; Woo et al., 2018; Guo et al., 2022; Li et al., 2022). Yao et al. (2021) first suggested using an extra plug-and-play temporal-wise attention module for SNNs to bypass a few unnecessary input timesteps. Then they proposed a multi-dimensional attention module along temporal-wise, channel-wise, and spatial-wise separately or simultaneously to optimize membrane potentials, which in turn regulate the spiking response (Yao et al., 2023c). STSC-SNN (Yu et al., 2022) employed temporal convolution and attention mechanisms to improve spatio-temporal receptive fields of synaptic connections. SCTFA-SNN (Cai et al., 2023) computed channel-wise and spatial-wise attention separately to optimize membrane potentials along the temporal dimension. Yao et al. (2023a,b) recently proposed an advanced spatial attention module to harness SNNs' redundancy, which can adaptively optimize their membrane potential distribution by a pair of individual spatial attention sub-modules. TCJA-SNN (Zhu et al., 2022) cooperated temporal-wise joint channel-wise attention correlations using 1-D convolution operation. However, the temporal-channel receptive field of TCJA is a local cross shape that is restricted by its convolution kernels, requiring multiple repeated computations to establish long-range dependencies of features. Therefore, it is computationally inefficient and makes multi-hop dependency modeling difficult.

Among the attention mechanisms, self-attention, as another important feature of the human biological system, possesses the ability to capture feature dependencies. Originally developed for natural language processing (Vaswani et al., 2017), self-attention has been extended to computer vision, where it has achieved significant success in various applications. The self-attention module can also be considered a building block of CNN architectures, which are known for their limited scalability when it comes to large receptive fields (Han et al., 2022). In contrast to the progressive behavior of convolution operation, self-attention can capture long-range dependencies directly by computing interactions between any two positions, regardless of their positional distance. Moreover, it is commonly integrated into the top of the networks to enhance high-level semantic features for vision tasks. Recently, an emerging research direction is to explore the biological characteristics associated with the fusion of self-attention and SNNs (Yao et al., 2023a,b; Zhou C. et al., 2023; Zhou Z. et al., 2023). These efforts primarily revolve around optimizing the computation of self-attention within SNNs by circumventing multiplicative operations, leading to performance degradation. Diverging from these studies, our primary goal is to explore how self-attention can enhance the spatio-temporal information processing capabilities of SNNs.

2.2. Learning algorithms for SNNs

Existing SNN training methods can be roughly divided into three categories: 1) the biologically plausible method, 2) the conversion method, and 3) the gradient-based direct training method. The first one is based on biological plausible local learning rules, like spike timing dependent plasticity (STDP) (Diehl and Cook, 2015; Kheradpisheh et al., 2018) and ReSuMe (Ponulak and Kasinski, 2010), but achieving high performance for deep networks is challenging. The conversion method offers an alternative way to obtain high-performance SNNs by converting a well-trained ANN and mapping its parameters to an SNN with an equivalent architecture, where the firing rate of the SNN acts as ReLU activation (Cao et al., 2015; Rueckauer et al., 2017; Sengupta et al., 2019; Ding et al., 2021; Bu et al., 2022; Wu et al., 2023). Moreover, some works explored post-conversion fine-tuning of converted SNNs to reduce latency and increase accuracy (Rathi et al., 2020; Rathi and Roy, 2021; Wu et al., 2021). However, this method is not suitable for neuromorphic datasets. The gradient-based direct training methods primarily include voltage gradient-based (Zhang et al., 2020), timing gradient-based (Zhang et al., 2021), and activation gradient-based approaches. Among them, the activation gradient-based method demonstrates notable effectiveness when performing challenging tasks. This approach uses surrogate gradients to address the non-differentiable spike activity issue, allowing for error back-propagation through time (BPTT) to interface with gradient descent directly on SNNs for end-to-end training (Neftci et al., 2019; Wu et al., 2019; Yang et al., 2021; Zenke and Vogels, 2021). These efforts have shown strong potential in achieving high performance by exploiting spatio-temporal information. However, further research is required to determine how to make better use of spatio-temporal data and how to efficiently extract spatio-temporal features. This is what we want to contribute.

3. Materials and methods

In this section, we first present the representation of event streams and the adopted spiking neuron model and later propose our STCA module based on this neuron model. Finally, we introduce the training method adopted in this paper.

3.1. Representation of event streams

An event, e, encodes three pieces of information: the pixel location (x, y) of the event, the timestamp t' recording the time when the event is triggered, and the polarity of each single event $p \in \{-1, +1\}$ reflecting an increase or decrease of brightness via $+1/-1$. Formally, a set of events at the timestamp t'can be defined as:

$$\boldsymbol{E}_{t'} = \left\{\left[\boldsymbol{x}_k, \boldsymbol{y}_k, \boldsymbol{t}', \boldsymbol{p}_k\right]\right\}_{k=1}^{N} \tag{1}$$

Assume the spatial resolution is $h \times w$, the event set equals to the spike pattern tensor $S_{t'} \in R^{2\times h\times w}$ at the timestamp t'. However, processing these events one by one can be inefficient due to the limited amount of information contained in a single event. We follow the frame-based representation in SpikingJelly (Fang et al., 2020) that transforms event streams into high-rate frame sequences during preprocessing. Each frame includes many blank (zero) areas, and SNNs can skip the computation of the zero areas in each input frame (Roy et al., 2019), improving overall efficiency.

3.2. Spiking neural models

Spiking neuron in SNNs integrates synaptic inputs from the previous layer and the residual membrane potential into the latest membrane potential. The Parametric Leaky integrate-and-fire (PLIF) model can learn the synaptic weight and membrane time constant simultaneously, which can enhance the learning capabilities of SNNs (Fang et al., 2021). The subthreshold dynamics of the PLIF neuron is defined as:

$$\tau \frac{dV(t)}{dt} = -\left(V(t) - V_{rest}\right) + X(t) \tag{2}$$

where $V(t)$ indicates the membrane potential of the neuron at time t, τ is the membrane time constant that controls the decay of $V(t)$, $X(t)$ is the input collected from the presynaptic neurons and V_{rest} is the resting potential. When the membrane potential $V(t)$ exceeds the neuron threshold at time t, the neuron will emit a spike, and then the membrane potential goes back to a reset value V_{rest}. We set $V_{rest} = V_{reset} = 0$. The iterative representation of the PLIF model can be described as follows:

$$\begin{cases} H^{t,l} = V^{t-1,l} + \frac{1}{\tau}\left(-\left(V^{t-1,l} - V_{reset}\right) + X^{t,l}\right) \\ S^{t,l} = \Theta\left(H^{t,l} - V_{th}\right) \\ V^{t,l} = \left(1 - S^{t,l}\right)H^{t,l} + V_{reset}S^{t,l} \end{cases} \tag{3}$$

where superscripts t and l indicate the time step and layer index. To avoid confusion, we use $H^{t,l}$ and $V^{t,l}$ to represent the membrane potential after neuronal dynamics and after the trigger of a spike in layer l at time-step t, respectively. V_{th} is the firing threshold. $S^{t,l}$ is determined by $\Theta(x)$, the Heaviside step function that outputs 1 if $x \geq 0$ or 0 otherwise. The time constant $\tau = 1/k(a)$, $k(a)$ is a sigmoid function $1/(1+\exp(-a))$ with a trainable parameter a.

3.3. Self-attention-based temporal-channel joint attention module

The processing of temporal information in SNNs is generally attributed to spiking neurons because their dynamics naturally depend on the temporal dimension. However, the LIF neuron and its variants including the PLIF neuron, only sustain very weak temporal linkages. Additionally, event streams are inherently time-dependent therefore, it is necessary to establish spatial–temporal correlations to improve data utilization. The focus of this work is to model temporal-wise and channel-wise attention correlations globally by adopting a self-attention mechanism. We present our idea of attention with a pluggable module termed the Self-attention-based Temporal-Channel joint Attention (STCA), which is depicted in Figure 2.

Formally, we collect intermediate the spatial feature of l-th layer at all time-steps $X^l = [\cdots, X^{t,l}, \cdots] \in \mathrm{R}^{T \times C \times H \times W}$ as the input of STCA module, where T is time-step, C denotes channels, H and W are height and width of the feature, respectively. The spatial feature $X^{t,l}$ can be extracted from the original input $S^{t,l}$:

$$X^{t,l} = \mathbf{BN}\left(\mathbf{Conv}\left(\mathbf{W}^{l}, \mathbf{S}^{t,l-1}\right)\right) \tag{4}$$

where BN $(\cdot)$ and Conv $(\cdot)$ mean the batch normalization and convolutional operation, W^l is the weight matrix, $S^{t,l-1}$ $(l \neq 1)$ is a spike tensor that only contains 0 and 1, and $X^{t,l} \in R^{C_l \times H \times W_l}$. To simplify the notation, bias terms are omitted. BN is a default operation following the Conv, we also omit it in the rest of this paper. Since each spatial feature $X^{t,l}$ in X^l is time-dependent, our idea of attention is to utilize the temporal correlation of these features. It is well known that each channel of feature maps corresponds to a specific visual pattern. Our STCA module aims to determine 'when' to attend to 'what' are semantic attributes of the given input. For efficiency, STCA only focuses on temporal and channel modeling, the spatial information of the feature is aggregated by using both avg-pooling and max-pooling operations as follows:

$$R^{l} = \mathbf{AvgPool}\left(X^{l}\right) + \mathbf{MaxPool}\left(X^{l}\right) \tag{5}$$

where AvgPool $(\cdot)$ and MaxPool $(\cdot)$ represent the outputs of the avg-pooling and max-pooling layer respectively, $R^l \in \mathrm{R}^{T \times C}$. The generated different temporal-channel context descriptors, avg-pooled features and max-pooled features, are merged and then fed into a self-attention (SA) block. We follow the convention (Wang et al., 2018) to formulate the SA block, where the input feature in layer l is $R^l \in \mathrm{R}^{T \times C}$, and the output feature is generated as:

$$a_i^l = \frac{1}{C(r_i)} \sum_{\forall j} f\left(r_i, r_j\right) g\left(r_j\right) \tag{6}$$

where $r_i \in \mathrm{R}^{1 \times C}$ and $a_i \in \mathrm{R}^{1 \times C}$ indicate the i^{th} position of the input feature R^l and output feature A^l, respectively. Subscript j is the index that enumerates all positions along the temporal domain, i.e., i, $j \in [1, 2, \ldots, T]$, and a pairwise function $f(\cdot)$ computes a representing relationship between i and all j. The function $g(\cdot)$ computes a representation of the input signal at time-step j, and the response is normalized by a factor $C(r_i)$. We use a simple extension of the Gaussian function to compute the similarity in an embedding space, and the function $f(\cdot)$ can be formulated as:

$$f\left(r_i, r_j\right) = e^{\theta(r_i)\varnothing(r_j)^{T}} \tag{7}$$

where $\theta(\cdot)$ and $\phi(\cdot)$ can be any embedding layers. If we consider the $\theta(\cdot)$, $\phi(\cdot)$, $g(\cdot)$ in the form of linear embedding: $\theta(R^l) = R^l W_\theta$, $\phi(R^l) = R^l W_\phi$, $g(R^l) = R^l W_g$, where $W_\theta \in \mathrm{R}^{C \times C_k}$, $W_\phi \in \mathrm{R}^{C \times C_k}$, $W_g \in \mathrm{R}^{C \times C_k}$, and set the normalization factor as $C(r_i) = \sum_{\forall j} f\left(r_i, r_j\right)$, the Eq. 6 can be rewritten as:

$$a_i^l = \frac{e^{r_i w_{\theta,i} w_{\varnothing,j}^{T} r_j^{T}}}{\sum_j e^{r_i w_{\theta,i} w_{\varnothing,j}^{T} r_j^{T}}} r_j w_{g,j} \tag{8}$$

where $w_{\theta,i} \in \mathrm{R}^{C \times 1}$ is the i^{th} row of the weight matrix W_θ. For a given index i, $\frac{1}{C(r_i)} f\left(r_i, r_j\right)$ becomes the softmax output along the dimension j. The formulation can be future rewritten as:

$$A^{l} = \mathbf{softmax}\left(R^{l} W_{\theta} W_{\varnothing}^{T} R^{l}\right) g\left(R^{l}\right) \tag{9}$$

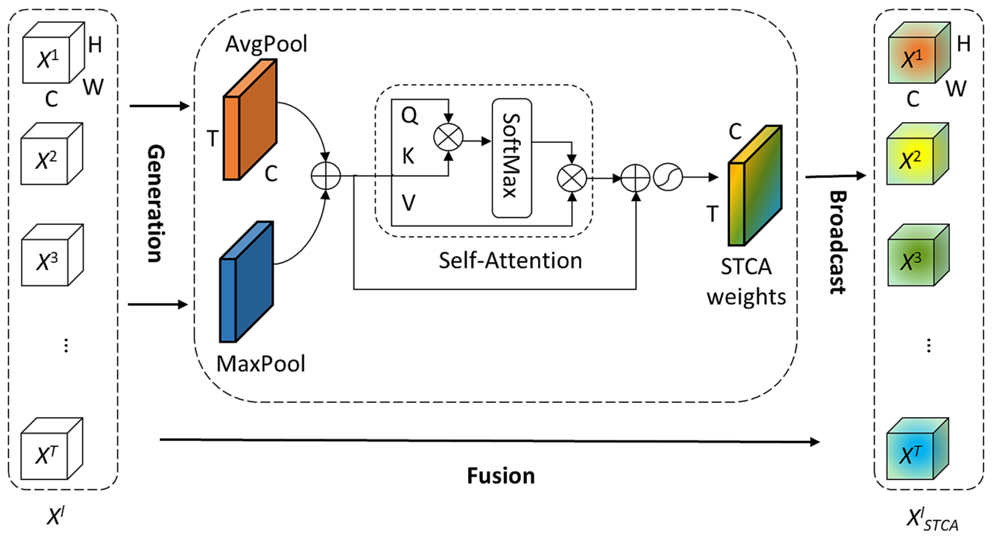

FIGURE 2
Diagram of the STCA module. The STCA module first aggregates spatial information by average-pooling and max-pooling then merges them and feeds it into a self-attention block to establish the correlations in both temporal and channel dimensions.

where $A^l \in R^{T\times C}$ is the output feature of the same size as R^l. Given the query, key, and value representations:

$$Q = R^l W^Q, K = R^l W^K, V = R^l W^V \qquad (10)$$

Once $W^Q = W_\theta, W^K = W_\phi, W^V = W_g$, $W^Q \in R^{C\times C}$, $W^K \in R^{C\times C}$, and $W^V \in R^{C\times C}$, Eq. 9 can be formulated as:

$$A^l = softmax\left(QK^T\right)V \qquad (11)$$

In this way, the SA block is constructed. Then we employ a residual connection around the SA block. Finally, the attention process of STCA can be formulated as:

$$X^l_{STCA} = f \odot X^l \qquad (12)$$

where $f = \sigma(R^l + A^l) \in R^{T\times C}$ is the weight vector of STCA, $\odot$ is element-wise multiplication, σ is the sigmoid function, and $X^l_{STCA} \in R^{T\times C\times H\times W}$ denotes the feature extracted by the STCA module along temporal and channel dimensions.

3.4. Training

We integrate the STCA module into networks and utilize the BPTT method to train SNNs. Since the process of neuron firing is non-differentiable, we use the derived ATan surrogate function $\sigma'(x) = \alpha / 2(1 + \pi\alpha x / 2)^2$. For a given input with label n, the neuron that represents class n has the highest excitatory level while other neurons remain silent. So the target output is defined by $Y = [y^{t,i}]$ with $y^{t,i} = 1$ for $i = n$, and $y^{t,i} = 0$ for $i \neq n$. Then the loss function is described by the spike mean squared error:

$$L = \left\| y^i - \frac{1}{T}\sum_{t=1}^{T} o^{t,i} \right\|^2 \qquad (13)$$

where $O = [o^{t,i}]$ is the average spiking events of neurons under the voting strategy.

4. Experiments

4.1. Experimental setup

4.1.1. Implementation details

We implement our experiments with the Pytorch package and SpikingJelly framework. All experiments were conducted using the BPTT learning algorithm on 4 NVIDIA RTX 2080 Ti GPUs. We utilized the Adam optimizer (Kingma and Ba, 2015) to accelerate the training process and implemented some standard training techniques of deep learning such as batch normalization and dropout. The corresponding hyper-parameters and SNN hyper-parameters are shown in Table 1. We verify our method on the following DVS benchmarks:

CIFAR10-DVS contains 10 K DVS images of 10 classes recorded with the dynamic vision sensor from the original static CIFAR10 dataset. We apply a 9: 1 train-valid split (i.e., 9 k training images and 1 k validation images). The resolution is 128 × 128, we resize all of them to 48 × 48 in our training and we integrate the event data into 10 frames per sample (Li et al., 2017).

N-Caltech 101 dataset contains 8,831 DVS images converted from the original version of Caltech 101 with a slight change in object classes to avoid confusion. The N-Caltech 101 consists of 100 object classes plus one background class. Similarly, we apply the 9: 1 train-test split as CIFAR10-DVS. We use the SpikingJelly (Fang et al., 2020) package to process the data and integrate them into 14 frames per sample (Orchard et al., 2015).

TABLE 1 Hyper-parameter setting.

Hyperparameter	N-MNIST	CIFAR10-DVS	N-Caltech 101
Max Epoch	500	1,000	500
Automatic mixed precision	×	×	✓
Batch size	64	32	8
Learning rate	1e-3	1e-3	1e-3
Time step	10	10	14
V_{th}	1.0	1.0	1.0
τ_0	2.0	2.0	2.0
head	4	4	4

TABLE 2 The network structures with STCA for different datasets.

Dataset	Network structure
N-MNIST	Input-128C3-Neuron-MP2-128C3-Neuron-STCA-MP2-0.5DP-2048FC-Neuron-0.5DP-100FC-Neuron-Voting
CIFAR10-DVS	Input-64C3-Neuron-128C3-Neuron-AP2-256C3-Neuron-256C3-Neuron-STCA-AP2-512C3-Neuron -512C3-Neuron-STCA-AP2-512C3-Neuron-512C3-Neuron-AP2-10FC-Neuron
N-Caltech 101	64C3-Neuron-MP2-128C3-Neuron-MP2-256C3-Neuron-STCA-MP2-256C3-Neuron-STCA-MP2-512C3-Neuron-0.8DP-1024FC-Neuron-0.5DP-101FC-Neuron

*x*C*y*/MP*y*/AP*y* denotes the Conv2D/MaxPooling/Avgpooling layer with output channel = *x*, and kernel size = $y \times y$, *n* FC denotes the fully connected layer with output feature = *n*, MP*y* is the spiking dropout layer with dropout ratio *m*. BN follows behind all *x*C*y*.

TABLE 3 Accuracy performance comparison between the proposed method and the SOTA methods on different datasets.

Method	Binary spikes	N-MNIST		CIFAR10-DVS		N-Caltech 101	
		T	Acc. (%)	T	Acc. (%)	T	Acc. (%)
tdBN (Yang et al., 2021)	✓	–	–	10	67.8	–	–
Rollout (Kugele et al., 2020)	✓	32	99.57	48	66.97	–	–
LIAF-Net (Wu et al., 2019)	×	20	99.13	10	70.4	–	–
ConvSNN (Samadzadeh et al., 2023)	✓	-	99.6	-	69.2	–	–
PLIF (Fang et al., 2021)	✓	10	99.61	20	74.80	–	–
TA-SNN (Yao et al., 2021)	×	–	–	10	72.0	–	–
SALT (Kim and Panda, 2021)	✓	–	–	20	67.1	20	55.0
STSC-SNN (Yu et al., 2022)	✓	10	99.64	10	81.4[a]	–	–
TCJA-SNN (Zhu et al., 2022)	✓	–	–	10	80.7[a]	14	78.5
This work	✓	10	99.67	10	81.6[a]	14	80.88

[a] With data augmentation.

The neuromorphic MNIST dataset is a converted dataset from the original static MNIST dataset (Orchard et al., 2015). It contains 50 K training images and 10 K validation images. We integrate the event data into 10 frames per sample using SpikingJelly (Fang et al., 2020) package.

4.1.2. Networks

The network structures with STCA for different datasets are provided in Table 2 and the network architectures we use have been proven to perform quite well on each dataset. Specifically, for the CIFAR10-DVS dataset, we adopt a VGG11-like architecture. To mitigate the apparent overfitting on the CIFAR10-DVS dataset, we adopt the neuromorphic data augmentation, including horizontal Flipping and Mixup in each frame, which is also used in Zhu et al. (2022) for training the same dataset. For the N-Caltech 101 dataset, we adopt the same architecture with Zhu et al. (2022) and N-MNIST refers to PLIF Fang et al. (2021). The voting layers are implemented using average pooling for classification robustness.

4.2. Comparison with existing state-of-the-art works

Table 3 displays the accuracy performance of the proposed STCA-SNNs compared to other competing methods on three neuromorphic datasets, N-MNIST, CIFAR10-DVS, and N-Caltech 101. We mainly include direct training results of SNNs with signal transmission via

TABLE 4 Accuracy of vanilla SNN, TA-SNN, and STCA-SNN models on different datasets.

Model	N-MNIST	CIFAR10-DVS	N-Caltech 101
Vanilla SNN	99.64	80.7	79.40
TA-SNN	99.64	81.3	80.76
STCA-SNN	99.67	81.6	80.88

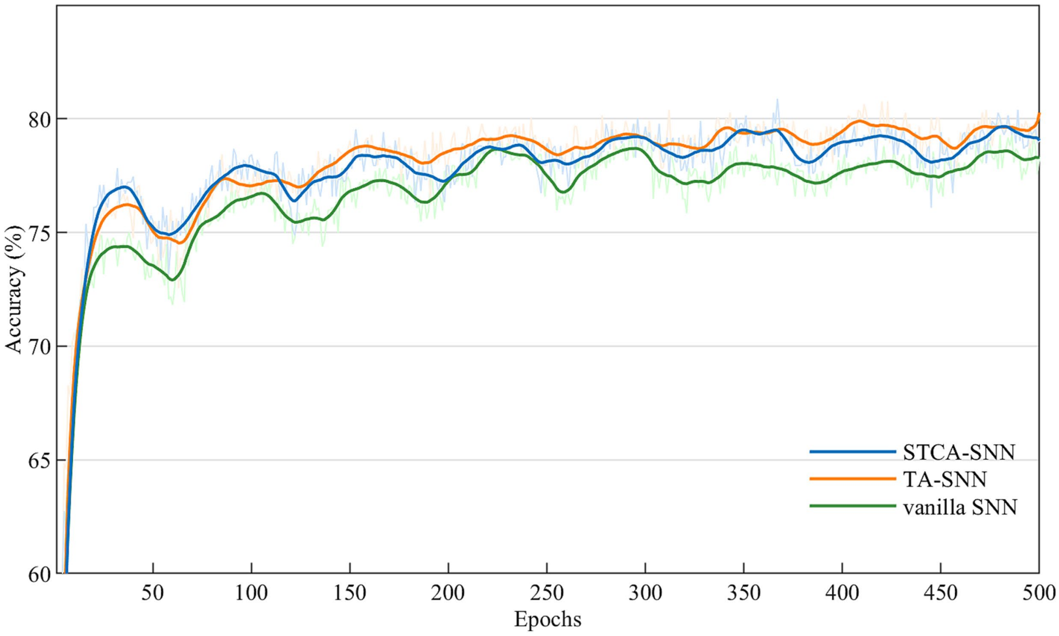

FIGURE 3
Convergence of compared SNN methods on N-Caltech101 dataset.

binary spike. Among them, some works (Wu et al., 2019; Yao et al., 2021) replace binary spikes with floating-point spikes and maintain the same forward pipeline as SNNs to obtain enhanced classification accuracy. STCA-SNNs achieve better performance than existing state-of-the-art SNNs on all datasets. We first compare our method on the CIFAR10-DVS dataset. We continue to utilize MSE the loss function and the same network architecture as TCJA-SNN (Zhu et al., 2022) and STSC-SNN (Yu et al., 2022) to preserve the consistency of this work, and our method reaches 81.6% top-1 accuracy, improving the accuracy by 0.9% over TCJA-SNN (Zhu et al., 2022). We also compare our method on N-Caltech 101dataset. Under the same condition as TCJA-SNN (Zhu et al., 2022) with MSE the loss function, we get a 2.38% increase over it and outperform the comparable result. Finally, we test our algorithm on the N-MNIST dataset. As shown in Table 3, most comparison works get over 99% accuracy. We use the same architecture as PLIF. Our STCA-SNN reaches the best accuracy of 99.67%.

4.3. Ablation study

4.3.1. Ablation study

We performed ablation experiments based on the PLIF neuron model to evaluate the effectiveness of the STCA module. For each dataset, we trained three types of SNNs: STCA-SNNs, TA-SNNs with temporal-wise attention module (Yao et al., 2023c), and vanilla SNNs (PLIF-SNN) without any attention module. The SE attention employed by TA-SNNs in the temporal dimension and the Self-attention employed in this work are both non-local operators, thus, we compared the performance of these two classic non-local operators under the same experiment conditions. We followed the learning process described in section 4.1 for all ablation experiments, and the attention locations were identical for both TA-SNNs and STCA-SNNs. Table 4 shows that all STCA-SNNs outperformed vanilla SNNs on three event stream classification datasets, suggesting that the benefits of the STCA module are not limited to a specific dataset or architecture. Furthermore, Figure 3 illustrates the accuracy performance trend of vanilla SNN, TA-SNN, and our proposed STCA-SNN over 1,000 epochs on the N-Caltech101 dataset. As the training epoch increased, our proposed STCA-SNN demonstrated comparable performance with TA-SNN. This indicates that our STCA module can enhance the representation ability of SNNs.

4.3.2. Discuss of pooling operations

To investigate the influence of the avg-pooling and max-pooling operation, we conducted several ablation studies. As is well known, avg-pooling can capture the degree information of target objects,

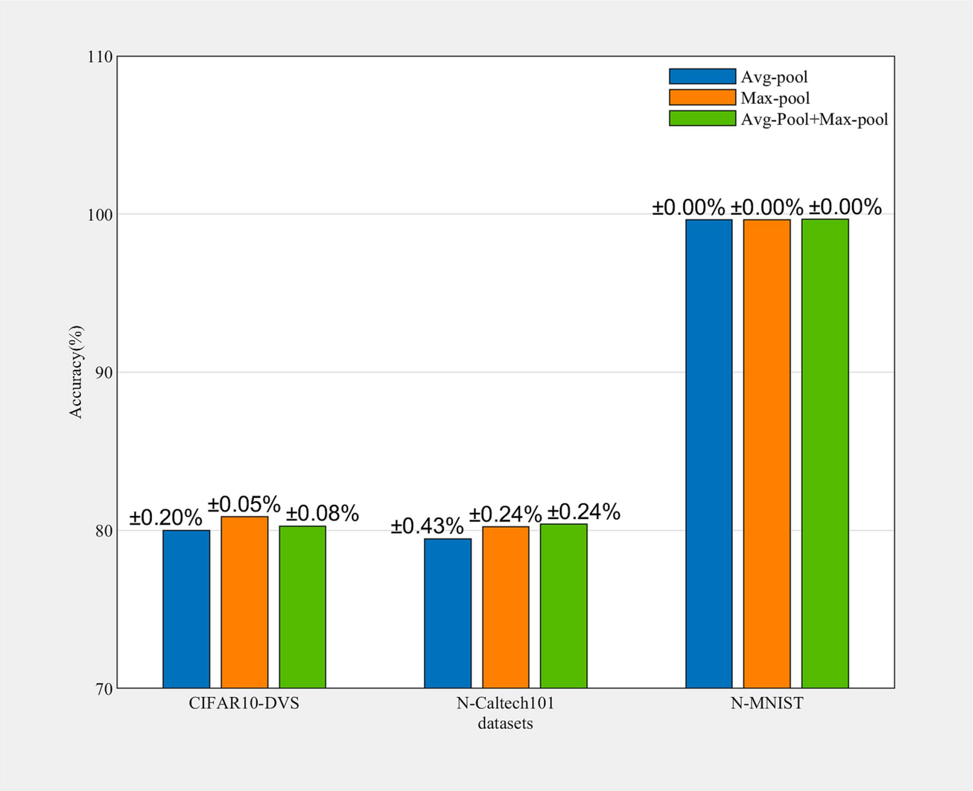

FIGURE 4
Accuracy of different datasets obtained by avg-pooling, max-pooling, and a combination of both. Each experiment is run 3 times.

while max-pooling can extract discriminative features of objects. As shown in Figure 4, the max-pooling operation contributes significantly to performance enhancement. Each experiment is run 3 times. Notably, the fusion of both pooling operations exhibits improved performance across all datasets examined, which means avg-pooling encoded global information can effectively compensate for the discriminative information encoded by max-pooling.

5. Conclusion

In this work, we propose the STCA-SNNs to enhance the temporal information processing capabilities of SNNs. The STCA module captures temporal dependencies across channels globally using self-attention, enabling the network to learn 'when' to attend to 'what'. We verified the performance of STCA-SNNs on various neuromorphic datasets across different architectures. The experimental results show that STCA-SNNs achieve competitive accuracy on N-MNIST, CIFAR10-DVS, and N-Caltech 101 datasets.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding authors.

Author contributions

XW: Conceptualization, Investigation, Methodology, Software, Visualization, Writing – original draft. YS: Funding acquisition, Supervision, Writing – review & editing. YZ: Supervision, Writing – review & editing. YJ: Supervision, Writing – review & editing. YB: Formal analysis, Validation, Writing – review & editing. XL: Formal analysis, Software, Validation, Visualization, Writing – review & editing. XY: Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported by the National Natural Science Foundation of China General Program (No. 82272130) and the National Natural Science Foundation of China Key Program (No. U22A20103).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., et al. (2017). A low power, fully event-based gesture recognition system. Proceedings of the IEEE conference on computer vision and pattern recognition.

Ba, J., Mnih, V., and Kavukcuoglu, K. (2014). Multiple object recognition with visual attention. In ICLR.

Bellec, G., Salaj, D., Subramoney, A., Legenstein, R., and Maass, W. (2018). Long short-term memory and learning-to-learn in networks of spiking neurons. 32nd conference on neural information processing systems.

Bu, T., Ding, J., Yu, Z., and Huang, T. (2022). Optimized potential initialization for low-latency spiking neural networks, the thirty-sixth AAAI conference on artificial intelligence (AAAI).

Cai, W., Sun, H., Liu, R., Cui, Y., Wang, J., Xia, Y., et al. (2023). A spatial-channel-temporal-fused attention for spiking neural networks. IEEE transactions on Neural Networks and Learning Systems. arXiv:2209.10837.

Cao, Y., Chen, Y., and Khosla, D. (2015). Spiking deep convolutional neural networks for energy-efficient object recognition. *Int. J. Comput. Vis.* 113, 54–66. doi: 10.1007/s11263-014-0788-3

Cheng, W., Luo, H., Yang, W., Yu, L., Chen, S., and Li, W. (2019). Det: a high-resolution dvs dataset for lane extraction. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pp. 1666–1675.

Diehl, P., and Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* 9:99. doi: 10.3389/fncom.2015.00099

Ding, J., Yu, Z., Tian, Y., and Huang, T. (2021). Optimal ANN-SNN conversion for fast and accurate inference in deep spiking neural networks. International joint conference on artificial intelligence.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). "An image is worth 16x16 words: transformers for image recognition at scale" in *International conference on learning representations (ICLR)*

Fang, W., Chen, Y., Ding, J., Chen, D., Yu, Z., Zhou, H., et al. (2020). Spikingjelly. Available at: https://github.com/fangwei123456/spikingjelly.

Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., and Tian, Y. (2021). Incorporating learnable membrane time constant to enhance learning of spiking neural networks. Proceedings of the IEEE/CVF international conference on computer vision, 2661–2671.

Gallego, G., Delbruck, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., et al. (2020). Event-based vision: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 44:1. doi: 10.1109/TPAMI.2020.3008413

Gerstner, W., Kistler, W. M., Naud, R., and Paninski, L. (2014). *Neuronal dynamics: From single neurons to networks and models of cognition*, Cambridge University Press, Cambridge, MA.

Guo, M. H., Xu, T. X., Liu, J. J., Liu, Z. N., Jiang, P. T., Mu, T. J., et al. (2022). Attention mechanisms in computer vision: a survey. *Comput. Visual Media* 8, 331–368. doi: 10.1007/s41095-022-0271-y

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., et al. (2022). A survey on vision transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 87–110. doi: 10.1109/TPAMI.2022.3152247

Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. Proceedings of the IEEE conference on computer vision and pattern recognition.

Hu, Y., Tang, H., and Pan, G. (2021). Spiking deep residual networks. *IEEETrans. Neural Netw. Learn. Syst.* 34, 5200–5205. doi: 10.1109/TNNLS.2021.3119238

Huang, Z., Zhang, S., Pan, L., Qing, Z., Tang, M., Liu, Z., et al. (2022). *TAda! oman. In ICLR.*

Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* 20, 1254–1259. doi: 10.1109/34.730558

Kheradpisheh, S., Mohammad, G., Thorpe, S. J., and Masquelier, T. (2018). STDP-based spiking deep convolutional neural networks for object recognition. *Neural Netw.* 99, 56–67. doi: 10.1016/j.neunet.2017.12.005

Kim, Y., and Panda, P. (2021). Optimizing deeper spiking neural networks for dynamic vision sensing. *Neural Netw.* 144, 686–698. doi: 10.1016/j.neunet.2021.09.022

Kingma, D. P., and Ba, J. L. (2015). Adam: a method for stochastic optimization. ICLR 2015: International conference on learning representations.

Kugele, A., Pfeil, T., Pfeiffer, M., and Chicca, E. (2020). Efficient processing of spatio-temporal data streams with spiking neural networks. *Front. Neurosci.* 14:439. doi: 10.3389/fnins.2020.00439

Li, G., Fang, Q., Zha, L., Gao, X., and Zheng, N. (2022). HAM: hybrid attention module in deep convolutional neural networks for image classification. *Pattern Recogn.* 129:108785. doi: 10.1016/j.patcog.2022.108785

Li, H., Liu, H., Ji, X., Li, G., and Shi, L. (2017). Cifar10-dvs: an event-stream dataset for object classification. *Front. Neurosci.* 11:309. doi: 10.3389/fnins.2017.00309

Lichtsteiner, P., Posch, C., and Delbruck, T. (2008). A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE J. Solid State Circuits* 43, 566–576. doi: 10.1109/JSSC.2007.914337

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). Swin transformer: hierarchical vision transformer using shifted windows. Proceedings of the IEEE/CVF International Conference on Computer Vision.

Mainen, Z. F., and Sejnowski, T. (1995). J, reliability of spike timing in neocortical neurons. *Science* 268, 1503–1506. doi: 10.1126/science.7770778

Neftci, E. O., Mostafa, H., and Zenke, F. (2019). Surrogate gradient learning in spiking neural networks: bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Process. Mag.* 36, 51–63. doi: 10.1109/MSP.2019.2931595

Orchard, G., Jayawant, A., Cohen, G. K., and Thakor, N. (2015). Converting static image datasets to spiking neuromorphic datasets using saccades. *Front. Neurosci.* 9:437. doi: 10.3389/fnins.2015.00437

Ponulak, F., and Kasinski, A. (2010). Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting. *Neural Comput.* 22, 467–510. doi: 10.1162/neco.2009.11-08-901

Posch, C., Matolin, D., and Wohlgenannt, R. (2010). A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds. *IEEE J. Solid State Circuits* 46, 259–275. doi: 10.1109/JSSC.2010.2085952

Rathi, N., and Roy, K. (2021). DIET-SNN: a low-latency spiking neural network with direct input encoding and leakage and threshold optimization. *IEEE Trans. Neural Networks Learn. Syst.* 34, 3174–3182. doi: 10.1109/TNNLS.2021.3111897

Rathi, N., Srinivasan, G., Panda, P., and Roy, K. (2020). Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation. International conference on learning representations.

Rebecq, H., Ranftl, R., Koltun, V., and Scaramuzza, D. (2019). High speed and high dynamic range video with an event camera. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 1964–1980. doi: 10.48550/arXiv.1906.07165

Ridwan, I., and Cheng, H., *An event-based optical flow algorithm for dynamic vision sensors* (2017) University of Lethbridge Lethbridge

Rieke, F., Warland, D., Van Steveninck, R. D. R., and Bialek, W. (1999). *Spikes: Exploring the neural code*. MIT Press, Cambridge, MA.

Roy, K., Jaiswal, A., and Panda, A. (2019). Towards spike-based machine intelligence with neuromorphic computing. *Nature*. 575, 607–617. doi: 10.1038/s41586-019-1677-2

Rueckauer, B., Lungu, I., Hu, Y., Pfeiffer, M., and Liu, S. (2017). Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Front. Neurosci.* 11:682. doi: 10.3389/fnins.2017.00682

Samadzadeh, A., Far, F. S. T., Javadi, A., Nickabadi, A., and Chehreghani, M. H. (2023). Convolutional spiking neural networks for spatio-temporal feature extraction. *Neural Processing Letters*. 1–7.

Sengupta, A., Ye, Y., Wang, R., Liu, C., and Roy, K. (2019). Going deeper in spiking neural networks: VGG and residual architectures. *Front. Neurosci.* 13:95. doi: 10.3389/fnins.2019.00095

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., et al. (2017). Attention is all you need. *Adv. Neural Inf. Proces. Syst.* 30, 5998–6008. doi: 10.48550/arXiv.1706.03762

Wang, X., Girshick, R., Gupta, A., and He, K. (2018). Non-local neural networks. Proceedings of the IEEE conference on computer vision and pattern recognition.

Woo, S., Park, J., Lee, J., and Kweon, I. (2018). Cbam: convolutional block attention module. Proceedings of the European conference on computer vision (ECCV).

Wu, Y., Deng, L., Li, G., Zhu, J., Xie, Y., and Shi, L. (2019). Direct training for spiking neural networks: Faster, larger, better, in Association for the Advancement of artificial intelligence (AAAI).

Wu, J., Xu, C., Han, X., Zhou, D., Zhang, M., Li, H., et al. (2021). Progressive tandem learning for pattern recognition with deep spiking neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 7824–7840. doi: 10.1109/TPAMI.2021.3114196

Wu, X., Zhao, Y., Song, Y., Jiang, Y., Bai, Y., Li, X., et al. (2023). Dynamic threshold integrate and fire neuron model for low latency spiking neural networks. *Neurocomputing* 544:126247. doi: 10.1016/j.neucom.2023.126247

Xu, Q., Qi, Y., Yu, H., Shen, J., Tang, H., Pan, G., et al. (2018). Csnn: an augmented spiking based framework with perceptron-inception. International Joint Conference on Artificial Intelligence (Stockholm).

Yang, Z., Wu, Y., Deng, L., Hu, Y., and Li, G. (2021). Going deeper with directly-trained larger spiking neural networks. *Neural Evol. Comput.* 35, 11062–11070. doi: 10.1609/aaai.v35i12.17320

Yao, M., Gao, H., Zhao, G., Wang, D., Lin, Y., Yang, Z., et al. (2021). Temporal-wise attention spiking neural networks for event streams classification. Proceedings of the IEEE/CVF international conference on computer vision (ICCV).

Yao, M., Hu, J., Zhao, G., Wang, Y., Zhang, Z., Xu, B., et al. (2023a). Inherent redundancy in spiking neural networks. Proceeding of the IEEE/CVF international conference on computer vision (ICCV). arXiv preprint arXiv:2308.08227.

Yao, M., Hu, J., Zhou, Z., Yuan, L., Tian, Y., Xu, B., et al. (2023b). Spike-driven Transformer. Advances in Neural Information Processing Systems (NeurIPS). arXiv preprint arXiv:2307.01694.

Yao, M., Zhao, R., Zhang, H., Hu, Y., Deng, L., Tian, Y., et al (2023c). Attention spiking neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 9393–9410. doi: 10.1109/TPAMI.2023.3241201

Yu, C., Gu, Z., Li, D., Wang, G., Wang, A., and Li, E. (2022). STSC-SNN: Spatio-temporal synaptic connection with temporal convolution and attention for spiking neural networks. *Front. Neurosci.* 16:1079357. doi: 10.3389/fnins.2022.1079357

Zenke, F., and Vogels, T. P. (2021). The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. *Neural Comput.* 33, 899–925. doi: 10.1162/neco_a_01367

Zhang, M., Luo, X., Chen, Y., Wu, J., Belatreche, A., Pan, Z., et al. (2020). An efficient threshold-driven aggregate-label learning algorithm for multimodal information processing. *IEEE J. Sel. Top Signal Process* 14, 592–602. doi: 10.1109/JSTSP.2020.2983547

Zhang, M., Wang, J., Wu, J., Belatreche, A., Amornpaisannon, B., Zhang, Z., et al. (2021). Rectified linear postsynaptic potential function for backpropagation in deep spiking neural networks. *IEEE Trans. Neural Netw. Learn Syst.* 33, 1947–1958. doi: 10.1109/TNNLS.2021.3110991

Zhou, C., Yu, L., Zhou, Z., Ma, Z., Zhang, H., Zhou, H., et al. (2023). Spikingformer: Spike-driven residual learning for transformer-based spiking neural network. arXiv preprint arXiv:2304.

Zhou, Z., Zhu, Y., He, C., Wang, Y., Yan, S., Tian, Y., et al. (2023). Spikformer: When spiking neural network meets transformer. ICLR, 2023. arXiv preprint arXiv:2209.15425.

Zhu, R., Zhao, Q., Zhang, T., Deng, H., Duan, Y., Zhang, M., et al. (2022). TCJA-SNN: Temporal-Channel joint attention for spiking neural networks. arXiv:2206.10177.