Spiking Neural Networks with Asynchronous Spatiotemporal Attention for Neuromorphic Vision

Yimeng Shan
Liaoning Technical University
Huludao, Liaoning, China
yimengshan2001@gmail.com

Haicheng Qu
Liaoning Technical University
Huludao, Liaoning, China
quhaicheng@lntu.edu.cn

Abstract

In recent years, the rapid expansion of the AI field has intensified the demand for edge computing, leading to increased attention on brain-inspired Spiking Neural Network (SNN) algorithms with inherent energy efficiency. Attention mechanisms have been repeatedly proven to significantly improve SNN performance, making the design of plug-and-play SNN attention modules a crucial component for advancing SNNs into the edge computing domain. However, current attention modules designed for SNNs often suffer from the inability to perform asynchronous computation in spatiotemporal dimensions, where this $O(T + L)$ complexity renders any acceleration designs on neuromorphic chips ineffective, hindering the progress of SNN algorithms toward edge computing. Therefore, we design a plug-and-play attention module, which performs attention across channel and temporal dimensions without compromising the asynchronous computation properties of SNNs in spatiotemporal dimensions. We also propose a regularization method based on the difference of attention weights, which effectively reduces the generalization error of SNN models by masking cross-confusion points between temporal and spatial dimensions. Experimental results on seven event-based datasets for classification, object detection and tracking tasks demonstrate that our method achieves state-of-the-art accuracy while preserving the asynchronous computation capabilities of SNNs in spatiotemporal dimensions, representing an important step toward high-performance SNN deployment at the edge.

Keywords: Spiking Neural Networks, Attention Mechanism, Neuromorphic Vision, Neuromorphic Computing.

1. Introduction

The rapid development of the AI industry presents new challenges for model deployment. In this context, brain-inspired spiking neural networks (SNNs), which offer the potential for energy-efficient computation, attract particular attention. SNN algorithms, combined with specially designed neuromorphic chips such as Loihi [28], achieve high-speed inference with minimal energy consumption. However, discrete spike signals limit the representation capability of SNNs, resulting in performance that typically falls below that of ANNs. Therefore, improving SNN performance constitutes a primary focus in current SNN algorithm research. Attention mechanisms, due to their plug-and-play nature and significant performance enhancement effects, emerge as important feasible solutions for training high-performance SNNs.

Plug-and-play attention modules dedicated to SNNs have been extensively studied. Due to the autoregressive nature of SNNs, research on SNN attention mechanisms inevitably focuses on the temporal dimension. Attention mechanisms focusing on the temporal dimension [51], as well as those combining temporal and spatial dimensions [64, 53, 48, 34, 36], have been widely investigated. Such investigations have even extended to different fusion approaches for spatiotemporal information [64] and multiscale spatiotemporal feature fusion methods [36]. However, existing studies suffer from several fundamental limitations: **(1)** They require multiple temporal inputs to be synchronously fed into the network and can only produce synchronous outputs, as illustrated in Fig. 1(a). **(2)** The computational units perform non-causal operations (shown as gray regions in Fig. 1(c)), which is impractical in real-world scenarios. **(3)** The substantial additional parameters and synchronous computations result in significantly higher training and inference latency, as demonstrated in Fig. 1(c). These limitations present considerable challenges for edge deployment of SNN algorithms.

Therefore, we propose Asynchronous Attention (Asyn-Attention), an asynchronous spiking spatiotemporal attention mechanism. SNNs incorporating Asyn-Attention can still directly produce predictions based on current temporal information given input at any time step. It consists of causal rules, simple Temporal-Attention (TA) and Channel-Attention (CA) components. By applying the strong con-

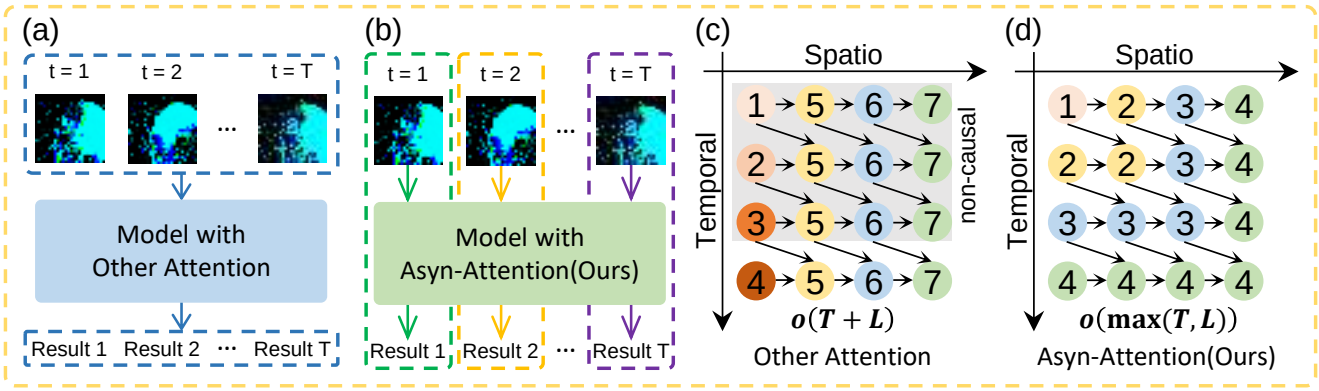

Figure 1: (a) and (b) respectively demonstrate the synchronous reasoning required by other attention mechanisms and the asynchronous reasoning of Asyn-Attention. (c) and (d) illustrate the training or inference costs of networks equipped with other attention mechanisms and Asyn-Attention, where the numbers within circles indicate computation completion time.

straint of causal computation rules—"**predictions at time t should only depend on inputs at or before time t**," Asyn-Attention achieves efficient asynchronous computation. CA comprises an Existence Branch (EB) and a Discriminative Branch (DB), where EB determines whether a channel has meaningful existence, and DB measures the importance of these channels. Additionally, we design an AttnOut regularization method that identifies spatiotemporal confusion points during training by leveraging the differential between channel attention branches and reusing temporal attention weights. By masking these confusion points for regularization, this method enhances the spatiotemporal correlation of SNNs while making the network more compact.

Event datasets possess inherent spatiotemporal characteristics. Therefore, we conduct comprehensive experiments on three event-based classification datasets, an event-based detection dataset and three event-based tracking datasets. Results demonstrate that our method achieves significant performance improvements with only negligible additional parameters, while maintaining faster inference speed compared to existing approaches. The main contributions of this paper are summarized as follows:

- We propose Asyn-Attention, the first attention module in SNNs that supports temporal causal operations, enabling asynchronous inference.
- We propose an AttnOut regularization method based on the differential between existence and discriminability and the reuse of attention weights.
- Extensive experiments demonstrate that our method achieves significant performance improvements with faster inference speed while requiring negligible parameter overhead. More importantly, the results prove that temporal attention maintains substantial effectiveness without non-causal computation.

2. Related work

2.1. Spiking Neural Networks

Brain-inspired SNNs possess significant potential for edge computing but cannot be directly trained through backpropagation due to non-differentiable spike signals. While ANN-to-SNN conversion methods [55] exist, they require extensive inference time steps and cannot handle neuromorphic datasets [5]. Direct training algorithms using surrogate gradient [23] and STBP [49] overcome these limitations, enabling efficient inference and effective spatiotemporal learning on neuromorphic datasets. These approaches have demonstrated success across neuromorphic classification [57, 30, 31, 43], object detection [40, 44] and event-based tracking [35, 39]. In this study, we train SNNs with Asyn-Attention using direct training algorithms and explore their asynchronous inference capabilities.

2.2. Attention Mechanism

Unlike ANNs [46, 47], due to the autoregressive temporal dynamics of SNNs, attention mechanisms in SNNs primarily operate along the temporal dimension. Early works propose temporal attention [51] and extend it to spatial and channel dimensions [53]. Subsequent developments include inhibitory attention with neuronal computation [34], channel-temporal correlation modules [64], cross-receptive field solutions [48] and multiscale spatiotemporal attention [36]. However, these approaches assume synchronous multi-time step data, making them impractical for real applications where networks cannot observe future time steps during inference, leading to temporal delays. Our proposed Asyn-Attention addresses this limitation, reducing inference time complexity from $O(T + L)$ to $O(\max(T, L))$ while achieving state-of-the-art accuracy.

3. Method

Following many existing works [36, 64], we employ the simplest architecture—Spiking VGG8, to validate the effectiveness of Asyn-Attention and AttnOut. We integrate Asyn-Attention into the fourth VGG-Block containing high-level features and identify spatiotemporal confusion points based on spatiotemporal attention weights for masking to facilitate spatiotemporal information flow, as illustrated in Fig. 2. Notably, we employ the simplest attention computation unit, as the primary innovation of our method does not lie in designing complex attention modules, but rather focuses on demonstrating that **attention modules can still achieve significant performance improvements even without utilizing non-causal operations**.

This section proceeds as follows: we first introduce LIF neurons [26] in Sec. 3.1, then analyze asynchronous issues in existing SNN attention mechanisms and propose our causal computation rule in Sec. 3.2. We subsequently present our Asyn-Attention mechanism in Sec. 3.3 and introduce AttnOut in Sec. 3.4.

3.1. Leaky Integrate-and-Fire neuron

In SNNs, the most widely used model is the Leaky Integrate-and-Fire (LIF) neuron [26], whose neural dynamics can be summarized as

$$\tau \frac{d\boldsymbol{V}(t)}{dt} = -(\boldsymbol{V}(t) - \boldsymbol{V}_{rest}) + \boldsymbol{I}(t), \tag{1}$$

where τ represents a time constant, $\boldsymbol{V}(t)$ denotes the membrane potential of the postsynaptic neuron, and $\boldsymbol{I}(t)$ signifies the input gathered from presynaptic neurons. Additionally, $\boldsymbol{V}_{rest}$ denotes the reset potential, which is established subsequent to activation of output spiking. To facilitate training and description, we adopt the displayed iteration version of the subthreshold dynamics model [27]

$$\boldsymbol{U}_t^n = \boldsymbol{H}_{t-1}^n + \frac{1}{\tau}(\boldsymbol{I}_{t-1}^n - (\boldsymbol{H}_{t-1}^n - \boldsymbol{U}_{rest})), \tag{2}$$

$$\boldsymbol{S}_t^n = \Theta(\boldsymbol{U}_t^n - \boldsymbol{U}_{threshold}), \tag{3}$$

$$\boldsymbol{H}_t^n = \boldsymbol{U}_t^n(1 - \boldsymbol{S}_t^n). \tag{4}$$

At each layer n and time step t, the membrane potential $\boldsymbol{U}$ of a neuron is denoted as $\boldsymbol{U}_t^n$. The parameter τ signifies a time constant, and $\boldsymbol{S}$ represents a binary spiking tensor. $\boldsymbol{I}$ denotes the neuron's input, while Θ represents the Heaviside step function. $\boldsymbol{H}$ symbolizes the hidden state, $\boldsymbol{U}_{rest}$ refers to the reset potential of neuron following a spike, and $\boldsymbol{U}_{threshold}$ indicates the discharge threshold of the neuron.

3.2. Asynchrony in Attention Spiking Neural Networks

The brain operates as a "massively asynchronous organ" where visual scene attribute integration occurs without a central temporal clock [3]. SNNs initially adhered to this neuroscientific principle, enabling asynchronous event-driven processing with theoretically unlimited scalability since each neuron processes inputs independently. However, attention mechanisms employing non-causal operations (using information from after time t to compute outputs at time t significantly improve performance but compromise asynchronicity. As shown in Fig. 1(c), the second layer cannot compute until the final time step information reaches the network, requiring waiting time equivalent to total time steps T. Despite subsequent parallel GPU computation, inference time remains prohibitive. This approach proves infeasible for real-time applications requiring moment-by-moment outputs, as attention-based SNN models need future inputs for attention weight calculation.

We achieve fully asynchronous SNNs by enforcing causal computation constraints: "**predictions at time t should only depend on inputs at or before time t.**" Under this constraint, as illustrated in Fig. 1(d), when information at time step t arrives, layer l simultaneously processes outputs from layer $l+1$ and new inputs at time step $t+1$. This enables layer-wise pipelined computation without waiting for complete temporal sequences, achieving two-fold inference efficiency improvement over non-asynchronous SNNs. The approach supports time step-by-time step inference with predictions at each time step, providing substantial efficiency gains for dense prediction scenarios. The time complexity of such asynchronous SNNs depends on the maximum of T and the number of network layers.

3.3. Asyn-Attention

Asyn-Attention operates across two dimensions: temporal and channel. The channel dimension is further divided into two branches based on their respective functions: the Existence Branch (EB) and the Discriminative Branch (DB), as illustrated in Fig. 2. Where EB determines whether a channel has meaningful existence, and DB measures the importance of these channels.

The Temporal Asyn-Attention (TA) is constructed based on the causal rule proposed in Sec. 3.2 and a simple attention computation module. For the input $X \in (C, H, W)$ at time step t, when $t = 1$, the system executes only

$$\text{memory}_t = \boldsymbol{X}, \text{memory} \in (\hat{T}, C, H, W), \tag{5}$$

where $\hat{T}$ represents the total time steps of the currently accumulated data in memory. When $t > 1$, the attention weights at the current time step are also computed through

$$\boldsymbol{W}_T^t = \text{TA}(\text{memory}), \tag{6}$$

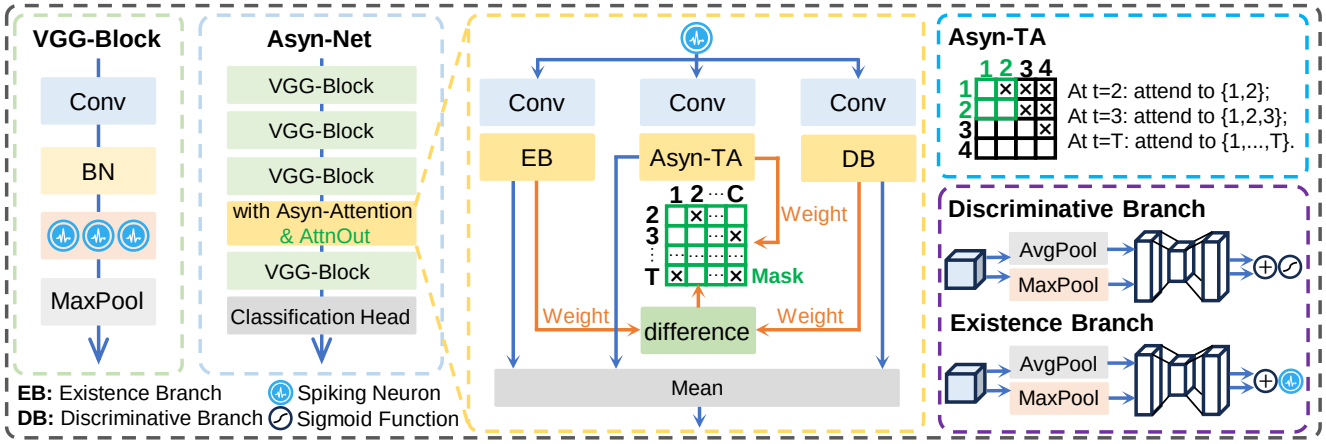

Figure 2: Asyn-VGG can be constructed by simply integrating Asyn-Attention into the VGG network. The implementation approach and overall concept of Asyn-Attention and AttnOut are illustrated in the figure.

TA first integrates importance and completeness information into the temporal dimension through maximum pooling and average pooling, respectively.

$$\boldsymbol{X}_{max} = \mathrm{MaxPool}(\boldsymbol{X}), \boldsymbol{X}_{max} \in (\hat{T}, 1), \tag{7}$$

$$\boldsymbol{X}_{avg} = \mathrm{AvgPool}(\boldsymbol{X}), \boldsymbol{X}_{avg} \in (\hat{T}, 1). \tag{8}$$

Subsequently, a Squeeze-and-Excitation (SE) module computes the attention weights for each, which can be formulated as

$$\mathrm{SE}(\boldsymbol{X}_{max}) = \mathrm{LN}^{\alpha}(\mathrm{LN}^{\beta}(\boldsymbol{X}_{max})), \tag{9}$$

$$\mathrm{SE}(\boldsymbol{X}_{avg}) = \mathrm{LN}^{\alpha}(\mathrm{LN}^{\beta}(\boldsymbol{X}_{avg})), \tag{10}$$

where LN represents linear operations, α and β represent the output channel numbers of the linear operations, and $\frac{\alpha}{\beta}$ constitutes the compression ratio of the SE module. In Sec. 4.2, we demonstrate through comprehensive ablation experiments that setting alpha to 2 achieves optimal performance in TA. For TA, beta varies dynamically with its value being $\hat{T}$.

The two branches are subsequently merged to obtain the final attention weights

$$\boldsymbol{W}_T = \mathrm{Sigmoid}(\mathrm{SE}(\boldsymbol{X}_{max}) + \mathrm{SE}(\boldsymbol{X}_{avg})), W_T \in (\hat{T}, 1). \tag{11}$$

Notably, although the attention weight computation at time step t incorporates all inputs from time steps $[1, t]$, only the corresponding output at time step t is applied to the Asyn-Attention input at that time step

$$\hat{\boldsymbol{X}} = \boldsymbol{X} \cdot \boldsymbol{W}_T^t, \hat{\boldsymbol{X}} \in (C, H, W). \tag{12}$$

Channel Attention (CA) in Asyn-Attention is constrained by causal rules. Based on storage from Eq. 6, the CA weights at time t can be calculated as

$$\boldsymbol{W}_C^t = \mathrm{CA}(\mathrm{memory}). \tag{13}$$

In CA, maximum pooling and average pooling are performed across both temporal and spatial dimensions, therefore the result can be expressed as

$$\boldsymbol{X}_{max} = \mathrm{MaxPool}(\boldsymbol{X}), \boldsymbol{X}_{max} \in (C, 1), \tag{14}$$

$$\boldsymbol{X}_{avg} = \mathrm{AvgPool}(\boldsymbol{X}), \boldsymbol{X}_{avg} \in (C, 1). \tag{15}$$

The remaining operations of the Discriminative Branch (DB) are identical to Eq. 9-12, but for the Existence Branch (EB), Eq. 11 is replaced by

$$\boldsymbol{W}_C^{EB} = \mathrm{LIF}(\mathrm{SE}(\boldsymbol{X}_{max}) + \mathrm{SE}(\boldsymbol{X}_{avg})), \boldsymbol{W}_C^{EB} \in (C, 1), \tag{16}$$

$\mathrm{SE}(\cdot)$ denotes a squeeze-and-excitation operation. LIF represents a single-layer LIF neuron with non-resetting membrane potential across time steps, following Shan et al. [34] who integrate LIF neurons into attention computation for inhibitory mechanisms. We deploy the DB and EB branches along the channel dimension to suppress ineffective noise and enhance attention to salient features. The resulting attention weights from both branches are applied to the current time step input, similar to W_T^t

$$Y = \hat{X} \cdot W_C^{EB} \cdot W_C^{DB}. \tag{17}$$

3.4. AttnOut

We utilize the temporal and channel attention weights obtained from the Asyn-Attention module to design an AttnOut regularization method with minimal computational overhead, as outlined in Algorithm 1.

In Algorithm 1, abs denotes the absolute value function. This differential operation between the discriminative branch and the existence branch essentially identifies confusion points along the channel dimension. Since each time

Algorithm 1: AttnOut

Input: Output after attention weight application : $\boldsymbol{Y}$; Current time step : $\hat{T}$; TA weights of Asyn-Attention : $\boldsymbol{W}_T$; EB weights of Asyn-Attention : $\boldsymbol{W}_C^{EB}$; DB weights of Asyn-Attention : $\boldsymbol{W}_C^{DB}$; Number of channels executing AttnOut : δ_c

Output: Y after AttnOut execution : $\boldsymbol{Z}$

$\boldsymbol{Z} = \boldsymbol{Y}$

if $\hat{T} = 1$ **then**

  return $\boldsymbol{Y}$

else

  Find the indices of the smallest value in array $\boldsymbol{W}_T : \boldsymbol{H} \in \mathbb{N}^{1\times 1}$

  $\boldsymbol{P} = \text{abs}(\boldsymbol{W}_C^{DB} - \boldsymbol{W}_C^{EB})$

  if $\hat{T} = H$ **then**

    Find the indices of the δ_c smallest values in array $\boldsymbol{P} : \boldsymbol{Q} \in \mathbb{N}^{\delta_c \times 1}$

  for *each* i *in* $\boldsymbol{Q}$ **do**

    $\boldsymbol{Z}_i = \boldsymbol{0}, \boldsymbol{Z} \in (C, H, W), \boldsymbol{Z}_i \in (H, W)$

  return $\boldsymbol{Z}$

step deemed weak undergoes this confusion point identification process, AttnOut inherently possesses spatiotemporal interaction. Furthermore, spatiotemporal confusion points introduce errors in the computational graph, and setting them to zero helps mitigate these errors to facilitate spatiotemporal information flow.

We formulate the training of networks equipped with AttnOut as a conditional constrained optimization problem. Let L_{task} denote the task-specific loss function, where the network training objective is to minimize $L_{task}(\boldsymbol{Y})$, with $\boldsymbol{Y}$ representing the network's predicted outputs. Through the computational graph connectivity, $\boldsymbol{Y}$ can be interpreted as feature representations from intermediate network layers. We impose the constraint that when $\hat{T} = H$, for all indices $i \in P(\hat{T})$, the condition $Y_i = \boldsymbol{0}$ holds.

Through Lagrangian duality theory, the aforementioned constrained optimization problem is equivalent to minimizing

$$L_{total}(\boldsymbol{Y}) = L_{task}(\boldsymbol{Y}) + \lambda \sum_{i \in P(\hat{T})} \|\boldsymbol{Y}_i\|_2^2 \cdot \mathbb{I}(\hat{T} = H), \tag{18}$$

where $\mathbb{I}(\cdot)$ is the indicator function that equals 1 if the condition holds and 0 otherwise. When $\lambda \to \infty$, any non-zero $\boldsymbol{Y}_i$ with $i \in P(\hat{T})$ renders the objective function unbounded, thus requiring the optimal solution to satisfy $\boldsymbol{Y}_i = 0$ for all $i \in P(\hat{T})$. This constraint precisely corresponds to the zeroing operation by AttnOut. This provides theoretical justification for the effectiveness of AttnOut, with the premise that smaller attention weights indeed correspond to lower importance of the respective time steps or channels. The basis for this assumption, beyond the evident principles of attention mechanisms, is further validated through experiments in Sec. 4.2.

4. Experiments

In Sec. 4.1 and Sec. 4.2, we present the public event datasets and experimental settings used in this work, respectively. In Sec. 4.3, we conduct ablation studies on the DVS128 Gesture [1] dataset to validate the effectiveness of each component in our proposed method and the attention position selection process. In Sec. 4.4, we compare our method with existing state-of-the-art approaches on event-based classification, object detection and tracking datasets to comprehensively demonstrate the effectiveness and generalizability of the proposed method.

4.1. Event-based Datasets

4.1.1 Event-based Classification Datasets

DVS128 Gesture [1] dataset is an event-based dataset that contains a sequence of 11 gestures. It contains 1,176 training samples and 288 test samples. The training and test sets were recorded by 23 and 6 subjects, respectively, under three lighting conditions.

CIFAR10-DVS [25] dataset is a conversion from the CIFAR10 dataset [22], consisting of ten categories totaling 10000 images. The size of each frame has also been expanded to 128×128 pixels. Due to the additional undesired artifacts included in the neuromorphic datasets obtained based on the conversion method, therefore, how to distinguish the categories of targets in complex backgrounds with additional errors is a challenging recognition task.

N-Caltech101 dataset [29] is captured by mounting an ATIS sensor on a motorized pan-tilt unit and moving the sensor while viewing Caltech101 [17] samples on an LCD monitor. It is a spiking version of the frame-based Caltech101 dataset. The original dataset contains "Faces" and "Faces Easy" classes, each consisting of different versions of the same images. To avoid confusion, the "Faces" class is removed from N-Caltech101, leaving 100 object classes and one background class. Each sample has a resolution up to 180×240, making it a highly challenging neuromorphic dataset.

4.1.2 Event-based Object Detection Datasets

Gen1 dataset [13] is designed for event-based object detection in automotive scenarios, where the primary detection targets are pedestrians and vehicles. It contains approximately 39.3 hours of real-world road driving recordings and approximately 255K manually annotated bounding

Table 1: Structures for Asyn-VGG.

Block	Asyn-VGG	Output Size		
		DVS128 Gesture	CIFAR10-DVS	N-Caltech101
1	3×3,64	128×128	128×128	180×240
	MaxPool(2,2,0,1)	64×64	64×64	90×120
2	3×3,128	32×32	32×32	45×60
	MaxPool(2,2,0,1)			
3	3×3,256	16×16	16×16	22×30
	MaxPool(2,2,0,1)			
4	3×3,512	8×8	8×8	11×15
	Asyn-Attention&AttnOut			
	MaxPool(2,2,0,1)			
5	3×3,512	4×4	4×4	5×7
	MaxPool(2,2,0,1)			
FC-1	AveragePool	1×1	1×1	1×1
	FC(2048)			
	Dropout(0.5)			
FC-2	FC(1024)	1×1	1×1	1×1
	Dropout(0.5)			
FC-3	FC(11/10/101)	1×1	1×1	1×1

boxes labeled at different frequencies for supervised learning and object detection evaluation. The dataset is captured using a GEN1/ATIS event camera mounted behind the vehicle windshield with a resolution of 304×240. The recordings are collected across multiple regions in France, covering diverse scenarios including urban, suburban, highway and rural environments with varying lighting and weather conditions. Individual video sequences range from tens of minutes to several hours. Gen1 represents the largest and most complex dataset currently available, providing the most comprehensive evaluation of event-based object detection capabilities.

4.1.3 Event-based Tracking Datasets

FE108 dataset [59] is collected using the DAVIS346 sensor [4], which integrates a dynamic vision sensor (DVS) with an active pixel sensor (APS) to capture synchronized event-based data and grayscale images at 346×260 pixels resolution. Ground-truth target localization is provided by a Vicon motion capture system with sub-millimeter accuracy at 240 Hz sampling frequency. Comprising 21 diverse target categories including animals, vehicles and everyday objects, this dataset exhibits event rates ranging from 0 to 3800 events/ms and is specifically designed to reflect challenging real-world conditions such as low illumination, motion blur and rapid target movements. As a benchmark particularly suited for evaluating event-based tracking and detection methods, FE108 emphasizes robustness in scenarios involving frequent target occlusion and loss during long-term tracking, making it an essential resource for advancing reliable event-based tracking algorithms through its high-quality annotations and rich scenario diversity.

FELT (First Event-based Long-Term Tracking) dataset [41] is specifically designed to address the challenges of long-term tracking under dynamic environmental conditions, emphasizing tracking stability and robustness over extended durations with particular focus on scenarios involving frequent target loss and subsequent recovery. Unlike conventional short-sequence datasets, FELT captures high-frequency event data that enables precise motion tracking even during rapid or subtle movements, presenting notable advantages over traditional frame-based datasets by providing detailed event-based motion information across prolonged sequences. This design facilitates thorough evaluation of long-term tracking algorithms in realistic and complex conditions, making it particularly beneficial for critical applications such as autonomous driving and surveillance where sustained tracking continuity is essential, thereby establishing FELT as a fundamental benchmark for advancing long-term tracking methodologies in event-based vision.

VisEvent dataset [42] is a comprehensive, large-scale resource specifically designed for event-based visual tasks in dynamic and realistic environments, featuring a broad spectrum of scenes and diverse object categories that establish a versatile foundation for evaluating event-based detection and tracking algorithms. As a leading benchmark in the event-based vision community, VisEvent enables researchers to develop and assess models capable of handling challenging conditions such as rapid movements, low illu-

Table 2: The comparison between the proposed methods and existing SOTA techniques on three mainstream neuromorphic classification datasets.

Work	DVS128		CIFAR10-DVS		N-Caltech101	
	T	Acc	T	Acc	T	Acc
PLIF [16]	20	97.6	20	74.8	-	-
tdBN [62]	40	96.9	10	67.8	-	-
SEW-ResNet [15]	16	97.9	16	74.4	-	-
HATS [37]	-	-	N/A	52.4	N/A	64.2
DART [32]	-	-	N/A	65.8	N/A	66.8
SALT [21]	-	-	20	67.1	20	55.0
TA-SNN [51]	60	98.6	10	72.0	-	-
TCJA-SNN [64]	20	**99.0**	10	80.7	14	78.5
STAA-SNN [61]	16	98.6	16	82.1	-	-
STCA-SNN [48]	-	-	10	81.6	14	80.88
MA-SNN [53]	20	98.3	-	-	-	-
Asyn-VGG(Ours)	**16**	**99.0**	**10**	**82.4**	**14**	**83.1**

mination and high dynamic ranges, while its structured design facilitates detailed analysis of how event-based methods address problems including occlusion and motion blur that traditional frame-based approaches often struggle with. By providing a robust experimental platform, VisEvent significantly contributes to advancing event-driven algorithms and positions itself as an indispensable resource for both academic research and practical implementations in event-based vision.

4.2. Implementation Details

For all classification, detection and tracking tasks, we conduct experiments using 4 $\times$ 4090 GPUs. In the classification task, the architecture of the Asyn-VGG network is shown in Tab. 1. All $\boldsymbol{U}_{threshold}$ in the network's LIF neurons are set to 1.0, τ is set to 2.0, and a hard reset strategy is employed. The Asyn-Attention and AttnOut strategies are applied exclusively to the fourth block, and ablation experiments concerning the selection of their application positions are presented in Sec. 4.3.

On the DVS128 Gesture dataset, we train for 200 epochs with a learning rate of 1e-4 and batch size of 9 using the AdamW optimizer. On the CIFAR10-DVS dataset, we train for 200 epochs with a learning rate of 1e-3 and batch size of 24 using the Adam optimizer. On the N-Caltech101 dataset, we train for 300 epochs with a learning rate of 1e-3 and batch size of 6 using the NAdam optimizer. All experiments employ the TET loss [14] function and apply cosine learning rate decay throughout training, without using any data augmentation or preprocessing.

For the event-based object detection task, we employ an SGD optimizer with momentum set to 0.9 and weight decay set to 5e-4. The model is trained for 50 epochs with a batch size of 128 and a learning rate of 1e-2. We adopt the default data augmentation strategy from YOLOv3 [33].

For the event-based tracking datasets, since our method relies on temporal information, we utilize the LIF neuron version of SDTrack. The Asyn-Attention and AttnOut modules are applied to every depthwise separable convolution within the convolutional components, as well as to the MLPs within the attention blocks, thereby enhancing spatiotemporal feature extraction. Following the same strategy as SDTrack, we fine-tune the model on the sample pair matching task based on pretrained weights from ImageNet. For the FE108 dataset, we train the model for 100 epochs using 60,000 sample pairs with a learning rate of 4e-4, which decays by a factor of 10 at epoch 80. For the VisEvent dataset, we train for 100 epochs using 30,000 sample pairs with a learning rate of 4e-4, which decays by a factor of 10 at epoch 80. For the FELT dataset, we train for 300 epochs using 60,000 sample pairs with a learning rate of 4e-4, which decays by a factor of 10 at epoch 240. Throughout the entire training process, no data augmentation or preprocessing is applied.

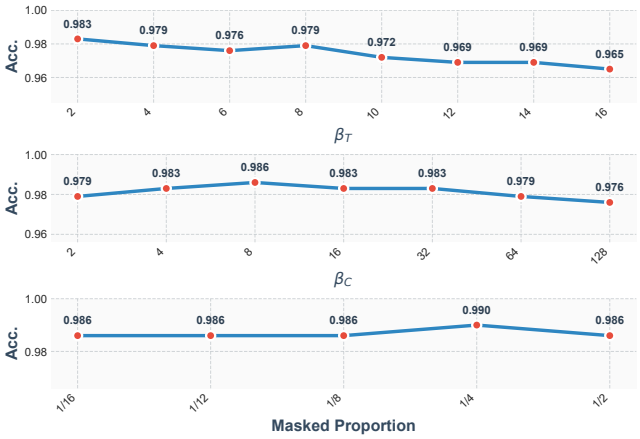

Figure 3: The upper figure demonstrates the impact of β_T on model accuracy when only TA is incorporated. Based on the β_T selected from the upper figure, the middle figure presents experiments on the impact of β_C on performance. The lower figure explores the optimal masked proportion of AttnOut based on Asyn-Attention with optimal parameters.

4.3. Ablation Study

We conduct extensive experiments on the DVS128 Gesture dataset to evaluate hyperparameter selection, application positions and component effectiveness for Asyn-Attention and AttnOut under identical settings.

Hyperparameter Selection. To ensure scale consistency, we constrain the beta values of the global pooling branch and average pooling branch in Asyn-Attention to be identical, thereby reducing the hyperparameters of Asyn-

Table 3: Comparison with standard pipeline on three event-based tracking benchmarks. The results of other methods are derived from the baseline established by SDTrack. In the table, 4×1 time steps denote the use of LIF neurons with $T = 4$, while 2×2 time steps denote the use of I-LIF neurons [52] with $T = 2$ nested within LIF neurons with $T = 2$.

Methods	Param. (M)	Time steps ($T \times D$)	FE108		FELT		VisEvent	
			AUC(%)	PR(%)	AUC(%)	PR(%)	AUC(%)	PR(%)
STARK [50]	28.23	1×1	57.4	89.2	39.3	50.8	34.1	46.8
SimTrack [7]	88.64	1×1	56.7	88.3	36.8	47.0	34.6	47.6
OSTrack_{256} [54]	92.52	1×1	54.6	87.1	35.9	45.5	32.7	46.4
ARTrack_{256} [45]	202.56	1×1	56.6	88.5	39.5	49.4	33.0	43.8
SeqTrack-B_{256} [8]	90.60	1×1	53.5	85.5	33.0	42.0	28.6	43.3
HiT-B [19]	42.22	1×1	55.9	88.5	38.5	48.9	34.6	47.6
GRM [18]	99.83	1×1	56.8	89.3	37.2	47.4	33.4	47.7
HIPTrack [6]	120.41	1×1	50.8	81.0	38.2	48.9	32.1	45.2
ODTrack [63]	92.83	1×1	43.2	69.7	29.7	35.9	24.7	34.7
*SiamRPN [24]	–	1×1	–	–	–	–	24.7	38.4
*ATOM [11]	–	1×1	–	–	22.3	28.4	28.6	47.4
*DiMP [2]	–	1×1	–	–	37.8	48.5	31.5	44.2
*PrDiMP [12]	–	1×1	–	–	34.9	44.5	32.2	46.9
*MixFormer [10]	37.55	1×1	–	–	38.9	50.4	–	–
*STNet [58]	20.55	3×1	–	–	–	–	35.0	50.3
*SNNTrack [60]	31.40	5×1	–	–	–	–	35.4	50.4
SDTrack [35]	19.61	4×1	56.7	89.1	35.8	44.0	35.4	48.7
		2×2	55.3	88.1	35.7	45.3	35.4	49.5
SDTrack + Asyn-Attention + AttnOut (Ours)	19.97	4×1	**58.8(+2.1)**	**90.9(+1.8)**	**39.7(+3.9)**	51.3(+7.3)	35.6(+0.2)	49.7(+1.0)
		2×2	57.6(+2.3)	90.6(+2.5)	38.6(+2.9)	**51.8(+6.5)**	**35.9(+0.5)**	**50.6(+1.1)**

Table 4: Accuracy under different insertion positions (single vs. pair).

Placement	baseline	1	2	3	4	5
Acc.(%)	96.9	97.2	97.9	98.3	**99.0**	98.6
Placement	1&2	1&3	2&3	3&4	4&5	1&5
Acc.(%)	96.9	97.9	98.3	98.6	**99.0**	97.9

Table 5: Incremental Testing.

Experiment	Acc.
Only TA	97.9
Only CA	97.2
TA + DB	98.3
TA + EB	98.3
TA + CA (Asyn-Attention)	**98.6**
Asyn-Attention + AttnOut-DB	98.3
Asyn-Attention + AttnOut-EB	98.6
Asyn-Attention + AttnOut-Different	**99.0**

Attention to two: β_T (the temporal dimension beta) and β_C (the channel dimension beta). AttnOut has one hyperparameter, which is the proportion of channels for masking. As shown in Fig. 3, we conduct ablation studies on the hyperparameters of these two algorithms and find that optimal performance is achieved when β_T is set to 2 and β_C is set to 8. For AttnOut, optimal performance is obtained when the masked proportion is set to 1/4 of the input channels.

Insertion Position Selection. Many works demonstrate that the insertion position and quantity of attention modules significantly impact performance. Therefore, we conduct experiments to investigate the optimal positions and quantities for Asyn-Attention and AttnOut. As shown in Tab. 4, applying them exclusively to the fourth block achieves the best performance.

Incremental Testing. Specifically, Asyn-Attention comprises three components: TA, DB and EB (where DB and EB constitute two branches of CA). The only component of AttnOut that supports incremental testing involves determining whether to utilize the attention weights from the DB branch, the EB branch, or their differential after applying masking at the current time step. We conduct comprehensive experiments addressing the aforementioned

scenarios, as presented in Tab. 5. The experimental results demonstrate that TA, DB and EB exhibit significant effectiveness when employed independently, and their combination simultaneously enhances model performance. Furthermore, utilizing the differential of DB and EB attention weights yields more substantial improvements in model generalization compared to the independent application of DB and EB attention weights. When TA and CA are employed simultaneously, the model performance significantly exceeds that achieved when using either component independently, thereby demonstrating that TA and CA exhibit synergistic effects.

4.4. Experimental Results on Neuromorphic Datasets

4.4.1 Event-based Classification

As shown in Tab. 2, we compare our method with existing state-of-the-art results on three neuromorphic classification datasets. The experimental results demonstrate that our approach achieves superior performance while maintaining fully causal computation. On the DVS128 Gesture dataset, our method achieves the same accuracy with T=16 as TCJA-SNN using T=20. On the CIFAR10-DVS and N-Caltech101 datasets, Asyn-VGG achieves the highest accuracy with the lowest simulation time steps. Notably, our method supports asynchronous computation and causal computation, which demonstrates that temporal attention in SNNs can achieve effective performance without accessing future input data.

Furthermore, we conduct inference time comparisons with representative attention works in SNNs, as shown in Tab. 6. To simulate realistic inference scenarios, we employ a Spiking VGG8 network backbone with identical attention modules under time step-by-time step inference mode. With 8 network layers and 16 simulation time steps, the time complexity of Asyn-SNN should theoretically be 2/3 that of other attention modules using non-causal operations, according to Fig. 1. However, actual results show an even smaller ratio, which we attribute to additional computational overhead from the complex designs in comparable works. Overall, our Asyn-Attention achieves faster inference than other SNN attention mechanisms, with actual latency closely matching the theoretical values.

4.4.2 Event-based Object Detection

The experimental results on the Gen1 dataset are presented in Tab. 7. By simply integrating Asyn-Attention and AttnOut into the first module with 512 channels in the ResNet backbone of EMS-YOLO, the performance achieves significant improvement. This fully demonstrates the effectiveness of our method in object detection tasks and its strong generalization capability on the Spiking ResNet architecture. Notably, our method introduces only 0.13M additional parameters.

Table 6: Inference Time Comparison with Other Attention.

Work	**Asyn(Ours)**	TA	TCJA	STAA	STCA	MA
Time	1×	1.57×	1.52×	1.68×	1.69×	1.74×

Table 7: Results on the Gen1 dataset.

Model	Param. (M)	T	mAP@ 50(%)	mAP@ 50:95(%)
YOLOv3-tiny [33]	10.2	1	44.5	-
EMS-YOLO [38]	6.2	5	54.7	26.7
	9.3	5	56.5	28.6
	14.4	5	59.0	31.0
VGG-11+SDD	12.6	1	-	17.4
MobileNet-64+SSD	24.3	1	-	14.7
DenseNet121-24+SSD [9]	8.2	1	-	18.9
Spiking-Yolo [20]	7.9	500	44.2	-
Tr-Spiking-Yolo [56]	7.9	5	45.3	-
EMS-YOLO	6.3(+0.13)	5	**55.1(+0.4)**	**26.9(+0.2)**
+Asyn-Attention	9.4(+0.13)	5	**56.9(+0.4)**	**28.9(+0.3)**
+AttnOut(Ours)	14.5(+0.13)	5	**59.7(+0.7)**	**31.4(+0.4)**

4.4.3 Event-based Tracking

The experimental results on event-based tracking tasks are shown in Tab. 3. Across three large-scale event-based tracking datasets, SDTrack achieves significant performance improvements by integrating Asyn-Attention and AttnOut, while introducing only negligible additional parameters. We attribute this substantial performance gain to the following mechanism: SDTrack aggregates inter-frame temporal information of event sequences into event frames through the event aggregation method GTP, and subsequently extracts and comprehends long-term temporal information within event frames via intra-frame autoregression of LIF neurons. However, this sequence-level temporal information and relational modeling relying on LIF neurons remains weak and insufficient to satisfy the demands of temporally-dependent event-based tracking tasks. In contrast, our Asyn-Attention strengthens this capability of the tracking model through learnable spatiotemporal attention weights, thereby yielding significant performance improvements. Additionally, event-based tracking tasks exhibit severe overfitting issues, and AttnOut provides effective mitigation of this overfitting problem.

5. Discussion

All ablation experiments are conducted on the classification task prior to comparative experiments. Therefore, we discuss the effectiveness of AttnOut on detection and tracking tasks here. As shown in Tab. 8, consistent with the

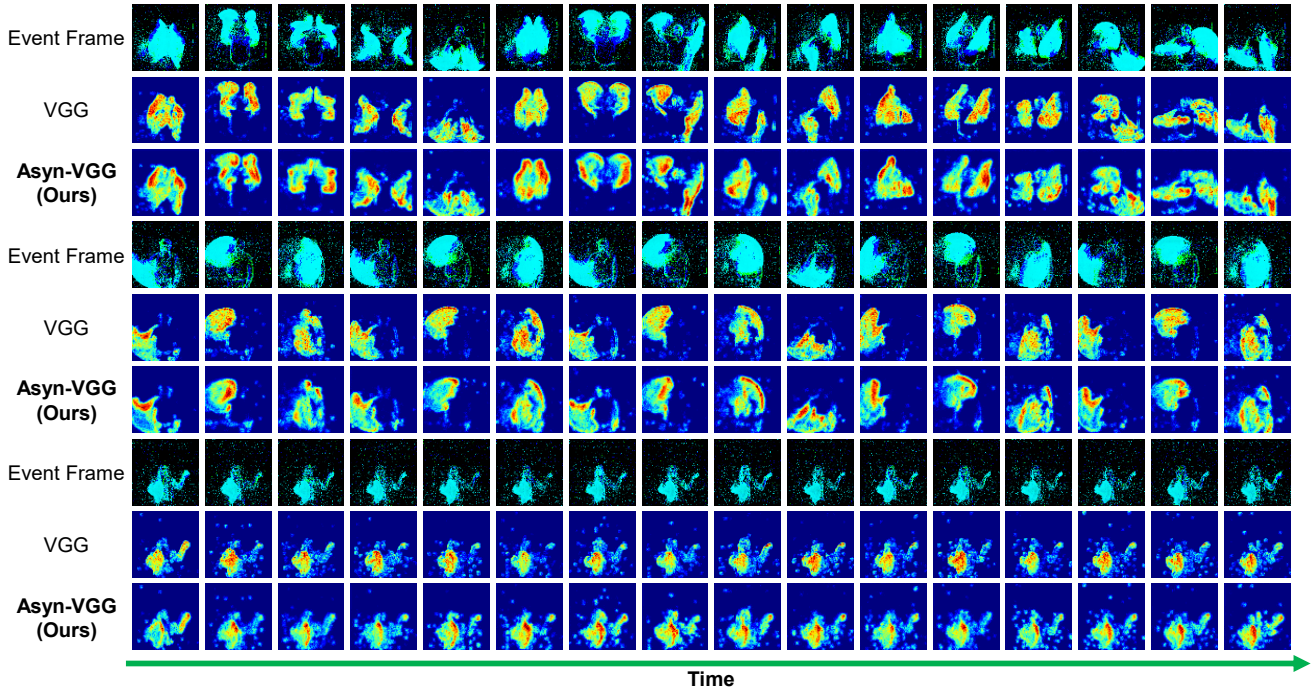

Figure 4: Visualization of typical sample input frames and their attentional heatmaps. The figure contains three groups of event sequences. The images with black backgrounds represent event frames corresponding to event streams, which serve as the network input. The second row displays heatmaps constructed from the firing rates of second-layer neurons in the baseline Spiking VGG8 network without any modifications. The third row presents heatmaps at corresponding locations for the VGG network augmented with Asyn-Attention and AttnOut modules, where red indicates high spiking activation and blue indicates low spiking activation.

Table 8: Additional Ablation Studies on Asyn-Attention and AttnOut for Event-based Detection and Tracking Tasks.

		Only Asyn-Attention		+AttnOut	
Task	Param.	mAP@ 50(%)	mAP@ 50:95(%)	mAP@ 50(%)	mAP@ 50:95(%)
Object Detection	6.3	54.9	26.8	55.1(+0.2)	26.9(+0.1)
	9.4	56.7	28.8	56.9(+0.2)	28.9(+0.1)
	14.5	59.4	31.2	59.7(+0.3)	31.4(+0.2)
Task	Dataset	AUC(%)	PR(%)	AUC(%)	PR(%)
Tracking	FE108	57.9	89.9	58.8(+0.9)	90.9(+1.0)
	FELT	39.4	50.9	39.7(+0.3)	51.3(+0.4)
	VisEvent	35.5	49.2	35.6(+0.1)	49.7(+0.5)

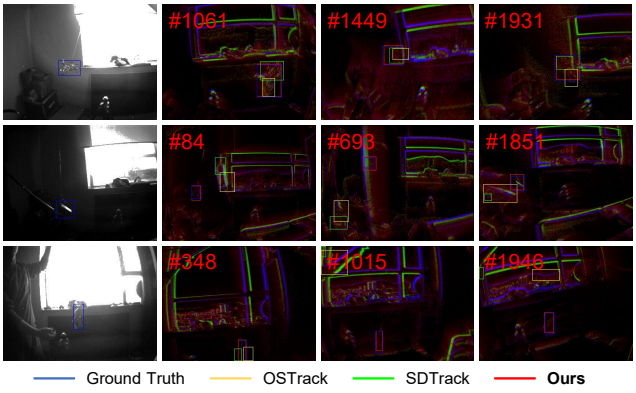

Figure 5: Tracking performance comparison between SDTrack with Asyn-Attention and AttnOut, Vanilla SDTrack and OSTrack. Our method demonstrates superior performance in multiple challenging scenarios and long-term tracking cases. Zoom in for better visualization.

event-based classification task, AttnOut significantly improves performance, particularly for the tracking task which suffers from severe overfitting.

We are also interested in understanding why Asyn-Attention and AttnOut are effective on event-based datasets. Thus, we perform qualitative analysis through visualization of intermediate feature maps. As shown in Fig. 4, the VGG network without Asyn-Attention tends to process each event frame as an independent static image. Not only is it difficult to observe traces of temporal information cap-

tured by LIF neurons across the event sequence, but the network's attention also remains at a coarse-grained level, failing to focus on critical information such as joints and edges. However, observation of the heatmaps generated from Asyn-VGG with Asyn-Attention and AttnOut reveals that it not only leverages temporal information between event frames (significant trajectory information is present in the third sample group) but also excels at capturing fine-grained features (arm edges and endpoints of arm motion). This fully demonstrates that the performance improvement of the network integrated with Asyn-Attention and AttnOut stems from enhanced capabilities in utilizing spatiotemporal information and capturing fine-grained features.

We further conduct visualization analysis on the more challenging event-based tracking task with longer event sequences. As shown in Fig. 5, in longer sequences, SDTrack faces the problem of target loss, whereas SDTrack integrated with Asyn-Attention does not encounter this issue. This is because the autoregression of LIF neurons is insufficient to capture the long-term temporal information that GTP methods integrate, but after integrating Asyn-Attention, the model's spatiotemporal feature extraction capability is enhanced, enabling it to capture trajectory clues of the target even in long-term tracking tasks.

6. Conclusion

In this study, we propose a causal constraint rule and introduce a plug-and-play Asyn-Attention module based on this. We further present an AttnOut method that reduces model generalization error by masking spatiotemporal confounding points. Experimental results on seven event-based classification, detection and tracking datasets demonstrate that our method achieves significant accuracy improvements while introducing negligible additional parameters. More importantly, we reduce the time complexity of temporal attention from $O(T+L)$ to $O(\max(T,L))$. Our approach represents an important step toward efficient SNN deployment. It also demonstrates that temporal attention without relying on impractical non-causal operations can still effectively improve SNN model performance on tasks dependent on temporal information.

References

[1] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. 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. 5

[2] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6182–6191, 2019. 8

[3] P. Bonzon. Symbolic modeling of asynchronous neural dynamics reveals potential synchronous roots for the emergence of awareness. *Frontiers in computational neuroscience*, 13:1, 2019. 3

[4] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck. A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014. 6

[5] T. Bu, W. Fang, J. Ding, P. Dai, Z. Yu, and T. Huang. Optimal ann-snn conversion for high-accuracy and ultra-low-latency spiking neural networks. *arXiv preprint arXiv:2303.04347*, 2023. 2

[6] W. Cai, Q. Liu, and Y. Wang. Hiptrack: Visual tracking with historical prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19258–19267, 2024. 8

[7] B. Chen, P. Li, L. Bai, L. Qiao, Q. Shen, B. Li, W. Gan, W. Wu, and W. Ouyang. Backbone is all your need: A simplified architecture for visual object tracking. In *European Conference on Computer Vision*, pages 375–392. Springer, 2022. 8

[8] X. Chen, H. Peng, D. Wang, H. Lu, and H. Hu. Seqtrack: Sequence to sequence learning for visual object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14572–14581, 2023. 8

[9] L. Cordone, B. Miramond, and P. Thierion. Object detection with spiking neural networks on automotive event data. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022. 9

[10] Y. Cui, C. Jiang, L. Wang, and G. Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13608–13618, 2022. 8

[11] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. Atom: Accurate tracking by overlap maximization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4660–4669, 2019. 8

[12] M. Danelljan, L. V. Gool, and R. Timofte. Probabilistic regression for visual tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7183–7192, 2020. 8

[13] P. De Tournemire, D. Nitti, E. Perot, D. Migliore, and A. Sironi. A large scale event-based detection dataset for automotive. *arXiv preprint arXiv:2001.08499*, 2020. 5

[14] S. Deng, Y. Li, S. Zhang, and S. Gu. Temporal efficient training of spiking neural network via gradient re-weighting. *arXiv preprint arXiv:2202.11946*, 2022. 7

[15] W. Fang, Z. Yu, Y. Chen, T. Huang, T. Masquelier, and Y. Tian. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34:21056–21069, 2021. 7

[16] W. Fang, Z. Yu, Y. Chen, T. Masquelier, T. Huang, and Y. Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2661–2671, 2021. 7

[17] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004. 5

[18] S. Gao, C. Zhou, and J. Zhang. Generalized relation modeling for transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18686–18695, 2023. 8

[19] B. Kang, X. Chen, D. Wang, H. Peng, and H. Lu. Exploring lightweight hierarchical vision transformers for efficient visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9612–9621, 2023. 8

[20] S. Kim, S. Park, B. Na, and S. Yoon. Spiking-yolo: spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11270–11277, 2020. 9

[21] Y. Kim and P. Panda. Optimizing deeper spiking neural networks for dynamic vision sensing. *Neural Networks*, 144:686–698, 2021. 7

[22] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[23] C. Lee, S. S. Sarwar, P. Panda, G. Srinivasan, and K. Roy. Enabling spike-based backpropagation for training deep neural network architectures. *Frontiers in neuroscience*, 14:497482, 2020. 2

[24] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8971–8980, 2018. 8

[25] H. Li, H. Liu, X. Ji, G. Li, and L. Shi. Cifar10-dvs: an event-stream dataset for object classification. *Frontiers in neuroscience*, 11:244131, 2017. 5

[26] W. Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997. 3

[27] E. O. Neftci, H. Mostafa, and F. Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019. 3

[28] G. Orchard, E. P. Frady, D. B. D. Rubin, S. Sanborn, S. B. Shrestha, F. T. Sommer, and M. Davies. Efficient neuromorphic signal processing with loihi 2. In *2021 IEEE Workshop on Signal Processing Systems (SiPS)*, pages 254–259. IEEE, 2021. 1

[29] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:159859, 2015. 5

[30] X. Qiu, M. Zhang, J. Zhang, W. Wei, H. Cao, J. Guo, R.-J. Zhu, Y. Shan, Y. Yang, and H. Li. Quantized spike-driven transformer. *arXiv preprint arXiv:2501.13492*, 2025. 2

[31] H. Qu, M. Mu, and Y. Shan. Efficient classification method for hyperspectral images based on spiking neural network. *Journal of Applied Remote Sensing*, 18(3):036509–036509, 2024. 2

[32] B. Ramesh, H. Yang, G. Orchard, N. A. Le Thi, S. Zhang, and C. Xiang. Dart: distribution aware retinal transform for event-based cameras. *IEEE transactions on pattern analysis and machine intelligence*, 42(11):2767–2780, 2019. 7

[33] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 7, 9

[34] Y. Shan, X. Qiu, R.-j. Zhu, J. K. Eshraghian, M. Zhang, and H. Qu. Syna-resnet: Spike-driven resnet achieved through or residual connection. *arXiv preprint arXiv:2311.06570*, 2023. 1, 2, 4

[35] Y. Shan, Z. Ren, H. Wu, W. Wei, R.-J. Zhu, S. Wang, D. Zhang, Y. Xiao, J. Zhang, K. Shi, et al. Sdtrack: A baseline for event-based tracking via spiking neural networks. *arXiv preprint arXiv:2503.08703*, 2025. 2, 8

[36] Y. Shan, M. Zhang, R.-j. Zhu, X. Qiu, J. K. Eshraghian, and H. Qu. Advancing spiking neural networks towards multiscale spatiotemporal interaction learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 1501–1509, 2025. 1, 2, 3

[37] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1731–1740, 2018. 7

[38] Q. Su, Y. Chou, Y. Hu, J. Li, S. Mei, Z. Zhang, and G. Li. Deep directly-trained spiking neural networks for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6555–6565, 2023. 9

[39] S. Wang, M. Zhang, J. Wang, D. Zhang, Y. Shan, J. Zhang, Y. Xiao, H. Cao, H. Zhang, Z. Ma, et al. Bipolar self-attention for spiking transformers. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 2

[40] S. Wang, M. Zhang, D. Zhang, A. Belatreche, Y. Xiao, Y. Liang, Y. Shan, Q. Sun, E. Zhang, and Y. Yang. Spiking vision transformer with saccadic attention. *arXiv preprint arXiv:2502.12677*, 2025. 2

[41] X. Wang, J. Huang, S. Wang, C. Tang, B. Jiang, Y. Tian, J. Tang, and B. Luo. Long-term frame-event visual tracking: Benchmark dataset and baseline. *arXiv preprint arXiv:2403.05839*, 2024. 6

[42] X. Wang, J. Li, L. Zhu, Z. Zhang, Z. Chen, X. Li, Y. Wang, Y. Tian, and F. Wu. Visevent: Reliable object tracking via collaboration of frame and event flows. *IEEE Transactions on Cybernetics*, 2023. 6

[43] W. Wei, M. Zhang, J. Zhang, A. Belatreche, S. Wang, Y. Shan, H. Liu, H. Cao, G. Wang, Y. Yang, et al. S 2 nn: Sub-bit spiking neural networks. *arXiv preprint arXiv:2509.24266*, 2025. 2

[44] W. Wei, M. Zhang, Z. Zhou, A. Belatreche, Y. Shan, Y. Liang, H. Cao, J. Zhang, and Y. Yang. Qp-snn: Quantized and pruned spiking neural networks. *arXiv preprint arXiv:2502.05905*, 2025. 2

[45] X. Wei, Y. Bai, Y. Zheng, D. Shi, and Y. Gong. Autoregressive visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9697–9706, 2023. 8

[46] Y. Wei, H. Qu, Y. Shan, Y. Gao, and J. Li. Beyond static filters: Dynamic convolutional transformers for multilingual handwritten text recognition. *Digital Signal Processing*, page 105638, 2025. 2

[47] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 2

[48] X. Wu, Y. Song, Y. Zhou, Y. Jiang, Y. Bai, X. Li, and X. Yang. Stca-snn: self-attention-based temporal-channel joint attention for spiking neural networks. *Frontiers in Neuroscience*, 17:1261543, 2023. 1, 2, 7

[49] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018. 2

[50] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu. Learning spatio-temporal transformer for visual tracking. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10448–10457, 2021. 8

[51] M. Yao, H. Gao, G. Zhao, D. Wang, Y. Lin, Z. Yang, and G. Li. Temporal-wise attention spiking neural networks for event streams classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10221–10230, 2021. 1, 2, 7

[52] M. Yao, X. Qiu, T. Hu, J. Hu, Y. Chou, K. Tian, J. Liao, L. Leng, B. Xu, and G. Li. Scaling spike-driven transformer with efficient spike firing approximation training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 8

[53] M. Yao, G. Zhao, H. Zhang, Y. Hu, L. Deng, Y. Tian, B. Xu, and G. Li. Attention spiking neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 45(8):9393–9410, 2023. 1, 2, 7

[54] B. Ye, H. Chang, B. Ma, S. Shan, and X. Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *European Conference on Computer Vision*, pages 341–357. Springer, 2022. 8

[55] K. You, Z. Xu, C. Nie, Z. Deng, Q. Guo, X. Wang, and Z. He. Spikezip-tf: Conversion is all you need for transformer-based snn. *arXiv preprint arXiv:2406.03470*, 2024. 2

[56] M. Yuan, C. Zhang, Z. Wang, H. Liu, G. Pan, and H. Tang. Trainable spiking-yolo for low-latency and high-performance object detection. *Neural Networks*, 172:106092, 2024. 9

[57] D. Zhang, S. Wang, Y. Xiao, W. Wei, Y. Shan, M. Zhang, and Y. Yang. Memory-free and parallel computation for quantized spiking neural networks. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025. 2

[58] J. Zhang, B. Dong, H. Zhang, J. Ding, F. Heide, B. Yin, and X. Yang. Spiking transformers for event-based single object tracking. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 8801–8810, 2022. 8

[59] J. Zhang, X. Yang, Y. Fu, X. Wei, B. Yin, and B. Dong. Object tracking by jointly exploiting frame and event domain. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13043–13052, 2021. 6

[60] J. Zhang, M. Zhang, Y. Wang, Q. Liu, B. Yin, H. Li, and X. Yang. Spiking neural networks with adaptive membrane time constant for event-based tracking. *IEEE Transactions on Image Processing*, 2025. 8

[61] T. Zhang, K. Yu, X. Zhong, H. Wang, Q. Xu, and Q. Zhang. Staa-snn: Spatial-temporal attention aggregator for spiking neural networks. *arXiv preprint arXiv:2503.02689*, 2025. 7

[62] H. Zheng, Y. Wu, L. Deng, Y. Hu, and G. Li. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11062–11070, 2021. 7

[63] Y. Zheng, B. Zhong, Q. Liang, Z. Mo, S. Zhang, and X. Li. Odtrack: Online dense temporal token learning for visual tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 7588–7596, 2024. 8

[64] R.-J. Zhu, M. Zhang, Q. Zhao, H. Deng, Y. Duan, and L.-J. Deng. Tcja-snn: Temporal-channel joint attention for spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. 1, 2, 3, 7