Spiking Transformers for Event-based Single Object Tracking

Jiqing Zhang[1], Bo Dong[2,⋆], Haiwei Zhang[1], Jianchuan Ding[1], Felix Heide[2], Baocai Yin[1], Xin Yang[1,⋆]
[1]Dalian University of Technology, [2] Princeton University

Abstract

Event-based cameras bring a unique capability to tracking, being able to function in challenging real-world conditions as a direct result of their high temporal resolution and high dynamic range. These imagers capture events asynchronously that encode rich temporal and spatial information. However, effectively extracting this information from events remains an open challenge. In this work, we propose a spiking transformer network, STNet, for single object tracking. STNet dynamically extracts and fuses information from both temporal and spatial domains. In particular, the proposed architecture features a transformer module to provide global spatial information and a spiking neural network (SNN) module for extracting temporal cues. The spiking threshold of the SNN module is dynamically adjusted based on the statistical cues of the spatial information, which we find essential in providing robust SNN features. We fuse both feature branches dynamically with a novel cross-domain attention fusion algorithm. Extensive experiments on three event-based datasets, FE240hz, EED and VisEvent validate that the proposed STNet outperforms existing state-of-the-art methods in both tracking accuracy and speed with a significant margin. The code and pretrained models are at https://github.com/Jee-King/CVPR2022_STNet.

1. Introduction

Event-based cameras are bio-inspired sensors that offer attractive properties compared to conventional frame-based cameras: high temporal resolution (in the order of μs), high dynamic range (140 dB vs. 60 dB), low power consumption, and high pixel bandwidth (on the order of kHz) resulting in drastically reduced motion blur [18]. With these unique sensing capabilities, event-based cameras can robustly function in degraded conditions, such as low-light, fast motion, and high dynamic range scenes. Recently, event-based cameras have been proposed for object tracking tasks [2,7–9,24,36,38], especially in adverse conditions that conventional imagers struggle with.

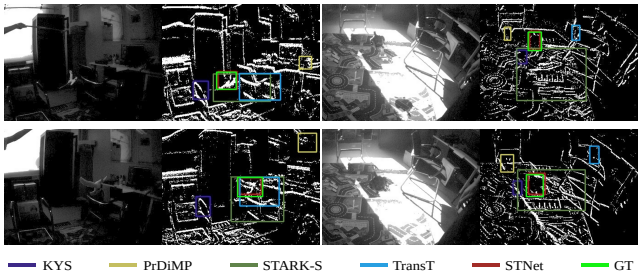


Figure 1. A comparison of our method (STNet) with state-of-the-art (SOTA) trackers. Unlike existing SOTA approaches, our method dynamically fuses temporal and spatial cues, resulting in robust tracking performance.

Event-based cameras measure per-pixel brightness changes and output events asynchronously. As a result, existing CNN-based approaches cannot directly work with event-based sensors, as they require synchronous input. Several works proposed to convert asynchronous events to conventional frames for downstream processing, mainly based on handcrafted features [27, 34, 41]. Unfortunately, in contrast to images captured with traditional cameras, the accumulated event frames are much sparser and lack texture information. Thus, directly applying CNN-based methods designed for conventional images does not offer a solution, as evidenced by extensive experiments in this work.

Recently, a line of works focused on developing learning-based object tracking approaches tailored to event frames. Zhang *et al*. [50] proposed to combine conventional and event frames with a cross-domain attention fusion algorithm to track objects under different degraded conditions. However, their approach requires both traditional and accumulated event frames as inputs. More importantly, the approach ignores the rich temporal information encoded in the event domain. In contrast, another body of work focused on the event domain only and introduced various event preprocessing approaches to encoding both temporal and spatial information into the processed event frames [8, 9, 30, 36]. However, these existing approaches do not consider the downstream network, making it challenging to choose suitable preprocessing methods for a specific computer vision task. Perhaps even more importantly, existing methods may suppress informative events from a downstream network

⋆ Xin Yang (xinyang@dlut.edu.cn) is the corresponding authors. Xin Yang and Bo Dong lead this project.

perspective.

In this work, we propose a spiking transformer network, dubbed STNet, for single object tracking, which only requires events as input. The proposed STNet does not require handcrafted event preprocessing but instead is designed to directly extract spatial and temporal information from the event positions over a small interval of time. The temporal information is extracted based on global spatial cues, and it plays an essential role in generalizing to different external conditions. The extracted temporal and spatial features are fused by a novel cross-domain attention method, which dynamically suppresses events based on the current scene.

In particular, the proposed architecture features a LIF-based spiking neural network (SNN) [45] for extracting temporal features and a reduced Swin-transformer [33] for extracting spatial information. We find it critical to operating the SNN based on spatial cues for this to function as intended. Specifically, the SNN neurons maintain membrane potential, which is increased by accumulating incoming spikes and decreased based on a decay function in the time domain [1, 21, 23]. A spike is generated when the accumulated potential is higher than a spiking threshold, and the potential is reset based on a refractory function [26, 42]. The potential accumulation, decay, and resting process can be regarded as a temporal memory, motivating us to treat the SNN as a temporal feature extractor. We ensure effective dynamic temporal features extraction by proposing to assign spiking thresholds to the LIF neurons based on the statistical cues of the spatial information.

Extensive experiments on the FE240hz [50], EED [36], and VisEvent [16] datasets validate the effectiveness of the proposed STNet (see Figure 1), which outperforms existing state-of-the-art methods by significant margins in representative success rate (RSR), representative precision rate (RPR), and processing speed (see Figure 7). Ablation experiments evidence the importance of each key component of STNet. More importantly, we show the spatial-aware dynamic spiking threshold is essential for robust tracking. Our work is the first work to verify the importance of the SNN dynamic threshold in object tracking tasks.

In summary, we make the following contributions:

- We propose a novel spiking transformer architecture for event-based single object tracking, allowing us to extract and fuse temporal and spatial information based on the dynamically defined informativeness of both temporal and spatial domains.
- We dynamically adjust the spiking threshold of a LIF-based SNN according to statistical cues of global spatial scene information.
- Extensively experimental results validate that the proposed approach outperforms state-of-the-art methods. Our ablation study evidences the effectiveness of each key component of the proposed STNet.

2. Background and Related Work

Event-based Cameras An event-based camera is a bio-inspired sensor that reports per-pixel brightness changes in log scale as a stream of asynchronous events [18, 20]. Compared to frame-based conventional cameras, event-based cameras offer a very high dynamic range (140 dB versus 60 dB) and high temporal resolution (in the order of μs). An event, e, encodes three pieces of information: the pixel location, (x, y), of the event, the timestamp, t, records the time when the event is triggered, and the polarity, $p \in \{-1, 1\}$, of an event, which reflects the direction of the changes. Formally, a set of events can be defined as

$$\mathcal{E} = \{e_k\}_{k=1}^{N} = \{[x_k, y_k, t_k, p_k]\}_{k=1}^{N}. \quad (1)$$

In constant lighting conditions, events are triggered by moving edges (*e.g.*, object contour and texture boundaries), making an event-based camera a natural edge extractor. However, this attractive feature is also a unique challenge. As the events predominately stem from edges, the measured events are inherently sparse. CNN-based approaches designed to work with conventional frames cannot work effectively with asynchronous and sparse events.

Spiking Neural Networks A spiking neural network (SNN) models biological information processing, in which neurons exchange information via spikes. Existing spiking neuron models mathematically describe the properties of a cell in the nervous system with varying degrees of detail. Typically, these models take three conditions into account: rest, depolarization, and hyperpolarization. When a neuron is resting, it keeps a constant membrane potential. The change in membrane potential can be either a decrease or an increase from the resting potential. An increase in membrane potential is called depolarization. In contrast, hyperpolarization describes a reduction in membrane potential, making a cell less likely to generate an action potential. All inputs and outputs to a spiking neuron model are sequences of spikes, which are called *spike trains*. A spike train is defined as $s(t) = \Sigma_{t^{(f)} \in \mathcal{F}} \delta(t - t^{(f)})$, where $\mathcal{F}$ is the set of times of the individual spikes [42]. Typical spiking neuron models set the resting potential as 0. However, existing models achieve depolarization and hyperpolarization in different ways.

In the following, we review the *leaky integrate-and-fire (LIF)* [22] spiking neuron model. LIF models are a simplified variant of Spike Response Models (SRMs) [21]. We can define an n_l-layer feedforward SNN architecture with LIF neurons. Given N^l incoming spike trains at layer l, $s_i^l(t)$, the SNN forward propagation is mathematically defined as

$$
\begin{aligned}
v_i^{l+1}(t) &= \sum_{j=1}^{N^l} w_{ij} s_j^l(t) + \\
&\quad v_i^{l+1}(t-1) f_d\left(s_i^{l+1}(t-1)\right) + b_i^{l+1}, \\
s_i^{l+1}(t) &= f_s\left(v_i^{l+1}(t)\right), \\
f_d(s(t)) &= \begin{cases} D & s(t) = 0 \\ 0 & s(t) = 1, \end{cases}
\end{aligned} \tag{2}
$$

where w_{ij} is the synaptic weight between the j-th neuron on the l-th layer and the i-th neuron on the layer $l+1$; b_i^{l+1} is an adjustable bias; and D is a constant. The operator $f_s(\cdot)$ is a spike function defined as

$$
f_s(v) : v \to s, s(t) := s(t) + \delta(t - t^{(f+1)}), \tag{3}
$$

$$
t^{f+1} = \min\{t : v(t) = \Theta, t > t^{(f)}\}, \tag{4}
$$

where Θ is the membrane potential threshold which is static and the same to all neurons in the network.

2.1. Event-based Object Tracking

Deep-learning-based algorithms have shown great success in object tracking tasks with conventional frame-based cameras [3, 4, 6, 10, 11, 14, 19, 25, 28, 29, 37, 48, 51]. However, conventional frame-based cameras suffer from degraded conditions (*e.g.*, high dynamic range scene, motion blur). Thereby, it requires significant algorithmic effort to achieve optimal tracking performance under these adverse conditions. In contrast, event-based cameras can naturally cope with these degraded conditions and gain substantial attention from the object tracking community. A line of works relies on different clustering algorithms to group captured events into clusters, including Gaussian Mixture Models [38], mean-shift [2], and particle filters [24]. However, these methods involve handcrafted strategies, requiring different tedious tuning for different application scenarios.

Another line of work converts the captured raw events to a specially designed format to encode both temporal and spatial cues for downstream tracking algorithms, such as time-surface [8], adaptive time-surface [9], and time image [36] formats. However, these event formats do not consider the downstream network, which may suppress informative cues from a downstream network perspective. Recently, Zhang *et al.* [50] proposed a cross-domain attention object tracking method, which can robustly function in degraded conditions. However, their approach did not leverage temporal information and required intensity frames. By contrast, our approach does not rely on special event preprocessing or input from other domains but on a novel network to dynamically extract and fuse temporal and spatial information for robust object tracking.

3. Spiking Transformers

Exploiting events captured by an event-based camera for object tracking requires tackling the following two challenges: 1) The event domain provides rich temporal information but in an asynchronous manner. Existing deep-learning-based approaches convert these asynchronous events to conventional intensity images for extracting spatial features [8, 9, 31, 36]. However, during the process, the original temporal information is partially lost; 2) Even if temporal information can be extracted from the events, dynamically fusing temporal and spatial cues remains challenging. In this work, to cope with these challenges, we propose a spiking transformer network (STNet) for single object tracking in the event domain, consisting of two key components: SNNformer feature extractor (SFE) and temporal-spatial feature fusion (TSFF).

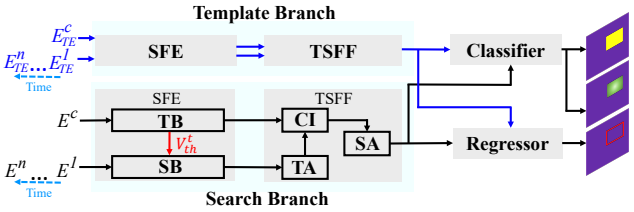


Figure 2. Overview of STNet. It has a template branch and a search branch, and they share the same architecture. Each branch contains a novel SNNformer Feature Extractor (SFE) and a novel Temporal-Spatial Feature Fusion (TSFF) module. The event-frames, $\{E_{TE}^c, E_{TE}^i\}$, are the inputs for template branch, and $\{E^c, E^i\}$ are the inputs for search branch, where $i \in [1, n]$.

The proposed SFE relies on a 3-layer LIF-based SNN [45] and a reduced Swin-transformer [33] to capture informative cues from the temporal and spatial domains. An SNN neuron accumulates its membrane potential from other directly connected neurons in a weighted sum manner, and the membrane potential is decayed based on different decay schemes [1, 21, 23]. When the accrued potential is higher than a spiking threshold, an action potential (*i.e.*, spike) is triggered, and then the membrane potential is rested based on various refractory strategies [26, 42]. This spiking mechanism acts as a natural noise filter. More importantly, we can regard the potential accumulation and decay as a temporal memory, which can effectively grasp temporal information. The original LIF's spiking threshold is pre-defined and fixed during training and testing. However, dynamic threshold schemes can be observed in the different biological nervous systems, which play an essential role in the adaptability to various external conditions [12, 17, 39, 43]. This observation inspires us to adjust the LIF threshold dynamically. We estimate a spiking threshold periodically based on the statistical cues of the spatial features, which are provided by a reduced Swin-transformer. As such, the proposed SFE can effectively counter the first key challenge from above.

We devise the fusion module TSFF to deal with the second challenge, which leverages three dynamic attention schemes to fuse the extracted temporal and spatial cues. Specifically, inspired by non-local self-attention [44], we

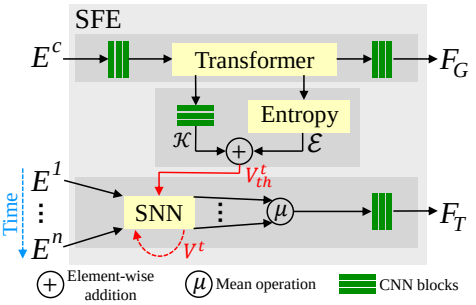


Figure 3. The architecture of the proposed SFE. Given a set of events of a time period, we generate E^c and E^i $i \in [1, n]$ based on Eq. 7 and Eq. 11, respectively. The SNN takes input event-frame one by one in time order and maintains its membrane potential V^t. The SFE dynamically changes the SNN threshold V_{th}^t.

propose a temporal-attention (TA) module to process the temporal features provided by the SNN inside the SFE, which utilizes channel-wise dependencies to enhance the temporal information. Next, a specifically designed cross-domain integrator (CI) combines enhanced temporal cues with spatial features. Essentially, CI generates an attention map based on spatial and temporal information to bridge the two domains. In doing so, low entropy information in the temporal domain can be future suppressed from the perspectives of both spatial and temporal domains. Finally, the fused features go through a spatial-attention (SA) module, where pixel-wise dependencies are leveraged to strengthen the discriminability of spatial features further.

For single object tracking, we adopt SiamFC++ [47] as our base network and replace the original frame-domain spatial feature extractors on both template and search branches with the proposed spiking transformer network. The proposed STNet is illustrated in Figure 2.

3.1. SNNformer Feature Extractor (SFE)

The proposed SFE has two branches: the transformer and SNN branches. In between, we offer a dynamic spiking threshold module for dynamically adjusting the SNN spiking threshold. We first introduce the input format and then give more details for each component. The overall architecture of the SFE is illustrated in Figure 3.

Event Input Format Given a set of events in a period, we split them into an n-bin voxel grid to discretize the time dimension. For each bin, we generate two frames, E_{neg} and E_{pos}, by recording the spatial positions of all occurred negative and positive events during the period, respectively.

Transformer Branch The transformer branch is used for extracting global spatial features from events. In contrast to frames captured by a conventional camera, the accumulated event frames are sparse and lack texture cues, which have been proved difficult for local feature extraction in today's CNN-based algorithms. We focus on global features and adopt the first two blocks of the Swin-transformer [33] as the feature extractor in the spatial domain. In particular, given n positive-negative pairs of accumulated event frames, the global spatial feature, F_G, is extracted as

$$F_G = \psi(\mathcal{C}_3(F_{\text{ST}})), \tag{5}$$

$$F_{\text{ST}} = \mathcal{T}(\mathcal{C}_1(E^c)), \tag{6}$$

$$E^c = [E^1_{pos}, E^2_{pos}, ..., E^n_{pos}, E^1_{neg}, E^2_{neg}, ..., E^n_{neg}], \tag{7}$$

where $[\cdot]$ is channel-wise concatenation; $\mathcal{C}_k$ denotes a convolutional layer with a kernel size of $k \times k$; $\mathcal{T}$ is the simplified Swin-transformer [33]; ψ is an operator consisting of Batch Normalization (BN) and a ReLU activation function.

SNN Branch The SNN network is based on the Spiking CNN [45] architecture, and it consists of three conv-SNN-blocks. In each block, a convolutional layer is followed by a LIF-based SNN layer, where the convolutional layer converts spikes to membrane potentials as input to the SNN layer. With n pairs of positive-negative event frames, the SNN branch sequentially processes them in time order. For each new input pair, the membrane potentials of all SNN neurons remain in their previous status instead of resetting to the initial potential status, which allows for capturing temporal cues. The membrane potentials are rest after processing all n pairs.

Mathematically, for a conv-SNN-based network, the membrane potentials, $V^{t,l}$, of the neurons in the l-th layer at the timestamp, t, is formulated as

$$V^{t,l} = H^{t-1,l} + \mathcal{C}\left(Z^{t,l-1}\right), \tag{8}$$

$$Z^{t,l} = f\left(V^{t,l} - \mathcal{V}_{th}\right), \tag{9}$$

$$H^{t,l} = \left(\alpha V^{t,l}\right)\left(1 - Z^{t,l}\right), \tag{10}$$

$$Z^{t,0} = E^t = \vee(E^t_{pos}, E^t_{neg}), \tag{11}$$

where $f(\cdot)$ is a Heaviside step function; $\mathcal{V}_{th}$ is the membrane potential threshold; α is the leakage factor of a LIF neuron; $\vee$ is an element-wise OR operator. At the last timestamp n, we calculate the mean of the last layer membrane potentials across all timestamps, from 1 to n, and use it to estimate the temporal features. The process is defined as

$$F_T = \psi(\mathcal{C}_1(\frac{1}{n}\sum_{t=1}^{n} V^{t,l=3})). \tag{12}$$

We note that the SNN with conv-SNN-blocks is different from the one with only LIF neurons. Hence, we cannot use the Eq. 2 to describe the conv-SNN-based network.

Dynamic Spiking Threshold The dynamic spiking potential threshold is a spontaneous regulation mechanism to inhibit over-excited (*i.e.*, high firing rate) and dead (*i.e.*, low firing rate) neurons. We adjust the spiking threshold of a LIF neuron based on spatial features of a scene. In particular, we use the representative value $\mathcal{K}$ and the entropy $\mathcal{E}$ of a given global spatial feature, F_G, to dynamically define a

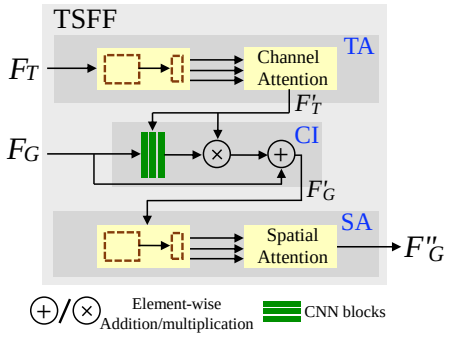


Figure 4. The architecture of the proposed TSFF. The module fuses extracted temporal and spatial features by specially designed channel and spatial attentions.

spiking threshold, $\mathcal{V}_{th}^t$ as

$$\mathcal{V}_{th}^t = (\mathcal{K} + \mathcal{E})\mathcal{V}_{th}, \tag{13}$$

$$\mathcal{K} = \sigma(\mathcal{A}(S_t^m)), \tag{14}$$

$$\mathcal{E} = -\frac{1}{N}\sum_{i=0}^{255}\frac{\mathcal{P}_i(\mathcal{N}(S_t^m))}{HW}\log(\frac{\mathcal{P}_i(\mathcal{N}(S_t^m))}{HW} + \xi), \tag{15}$$

where $\mathcal{V}_{th}$ is the pre-defined threshold, σ is a Sigmoid function, $\mathcal{A}$ is adaptive average pooling, S_t^m is a $H \times W$ matrix record the channel-wise means of the output features from the transformer, $\mathcal{N}$ is normalization operation, converting a value to the range of $[0, 255]$, $\mathcal{P}_i$ is the number of the pixels whose value is i on S_t^m; ξ is a numerical epsilon value.

Intuitively, $\mathcal{K}$ reflects the contextual information. The higher value of $\mathcal{K}$ reflects more contextual spatial information. In this case, we should reduce the neuron firing rate, allowing more accumulation and grasping temporal cues from a longer temporal window and vice versa. $\mathcal{E}$ reflects the randomness of the feature, which may be caused by spatial noise. For higher entropy observations, we want to increase the spiking threshold to filter out potential noise. In the opposite case, the spikes are more likely to be triggered by informative cues. Hence, we want to lower the threshold for increasing the sensitivity to these spikes.

3.2. Temporal-Spatial Feature Fusion (TSFF)

The proposed TSFF has three key components: Temporal-Attention (TA) module, Cross-Domain Integrator (CI), and Spatial-Attention (SA) module; see Figure 4.

Temporal-Attention (TA) Module Given the input temporal features, the proposed TA module utilizes channel-wise dependencies to produce the corresponding enhanced temporal cues, inspired by self-attention schemes [32, 35, 40, 44, 49]. Formally, given input temporal features, $F_T \in \mathbb{R}^{C\times H\times W}$, TA module is formulated as

$$F_T' = \mathcal{R}^{((C,H,W))}(\mathcal{G}Y) + F_T, \tag{16}$$

$$\mathcal{G} = \varphi(QK), \tag{17}$$

where $Y = Q = \mathcal{R}^{((C,HW))}(F_T), K = \mathcal{R}^{((HW,C))}(F_T)$; $\mathcal{R}^{((\cdot))}$ is reshape function with a target shape $(\cdot)$; φ denotes a softmax function.

Cross-Domain Integrator (CI) The CI is designed to fuse spatial and temporal cues with a cross-domain attention scheme. In particular, we generate an attention map based on both input spatial and temporal information and then apply it to the temporal cues. As such, we can suppress low-entropy information in the temporal domain based on the perspective of both spatial and temporal domains. Then, the CI fuses the input spatial and the optimized temporal features by element-wise addition. Given the input temporal features F_T', and spatial features F_G, this is

$$F_G' = \mathcal{X}F_T' + F_G, \tag{18}$$

$$\mathcal{X} = \sigma(\mathcal{M}(\mathcal{A}([F_G, F_T']))), \tag{19}$$

where $\mathcal{M}$ is a three-layer multilayer perceptron (MLP) operator with one linear input layer, one ReLU activation function, and one linear output layer.

Spatial-Attention (SA) Module The fused features are fed into a self-attention-based SA module to further enhance the discriminative spatial features. The SA module is similar to the TA module, but its attention map is based on pixel-wise dependencies rather than channel-wise dependencies. Formally, the SA module is

$$F_G'' = \mathcal{R}^{((C,H,W))}(Y'\mathcal{G}') + F_G', \tag{20}$$

$$\mathcal{G}' = \varphi(Q'K'), \tag{21}$$

where $Y' = \mathcal{R}^{((C,HW))}(F_G')$, $Q' = \mathcal{R}^{((HW,C))}(F_G')$, and $K' = Y'$.

3.3. Loss Function

To train the proposed STNet, we adopt the loss function of SiamFC++ [47], which contains three components: classification loss ($\mathcal{L}_{cls}$), cross-entropy loss ($\mathcal{L}_{ctr}$), and regression loss ($\mathcal{L}_{reg}$). The classification ground truth (GT), $c_{x,y}^*$, is set to 1 if the position (x, y) is a positive sample (*i.e.*, inside the GT bounding box), 0 otherwise. The cross-entropy loss relies on the GT $q_{x,y}^*$, which is obtained by setting a Gaussian function centered at the target bounding box. The loss function is defined as

$$\begin{aligned} L = &\frac{\lambda_1}{N_p}\sum_{x,y} L_{\text{cls}}\left(c_{x,y}, c_{x,y}^*\right) \\ &+ \frac{\lambda_2}{N_p}\sum_{x,y} f(c_{x,y}^*)L_{\text{ctr}}\left(q_{x,y}, q_{x,y}^*\right), \\ &+ \frac{\lambda_3}{N_p}\sum_{x,y} f(c_{x,y}^*)L_{\text{reg}}\left(t_{x,y}, t_{x,y}^*\right) \end{aligned} \tag{22}$$

where N_p is the number of positive samples; $f(\cdot)$ is the Heaviside step function; $t_{x,y}^*$ is the GT bounding box. λ_1, λ_2, and λ_3 are the weight factors, set empirically.

Implementation We implement the proposed network in PyTorch. The model is trained using a stochastic gradient descent (SGD) optimizer with a momentum of 0.9 and

a weight decay of $5e{-}5$. The STNet is trained for 20 epochs with batch size 38 on a 20-core i9-10900K 3.7 GHz CPU, 64 GB RAM, and an NVIDIA RTX3090 GPU. With this configuration, for each batch, STNet requires 26.9s for training. The learning rate is increased in $[1e{-}7, 2e{-}3]$ as the training progresses; for the first five epochs, it is linearly increased; for the rest 15 epochs, it is adjusted by a cosine annealed learning rate scheduler.

4. Experiments

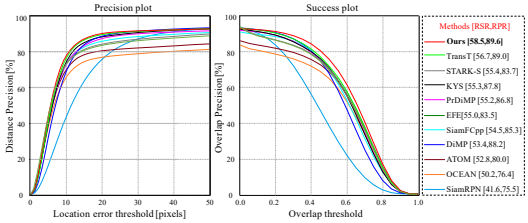
4.1. Experimental Setup

We use three datasets, FE240hz [50], EED [36], and VisEvent [16], for assessing the effectiveness of the proposed STNet. The FE240hz dataset is an extensive event-frame-based dataset for single object tracking, including more than 1132K annotations on more than 143K images and corresponding recorded events. It is captured under different degraded conditions (*e.g.*, motion blur, HDR) to cover diverse real-world scenarios. The EED dataset is a small dataset with only 199 bounding boxes, used only for validation purposes with STNet trained on FE240hz. As the FE240hz and EED datasets only contain rigid objects, we leverage the VisEvent dataset further to validate the effectiveness of our STNet with non-rigid objects. After removing sequences that miss event data or have misaligned timestamps, the VisEvent dataset includes 377 sequences for training and 172 for testing. Among the 172 testing sequences, 63 of them contain non-rigid objects. We preprocess event-frames as suggested by SiamFC++ [47] and use the following hyper-parameter settings: λ_1, λ_2, and λ_3 in Eq. 22 are set to 1, 1 and 3, respectively; 0.2 for the decay factor α in Eq. 10. The spiking threshold $\mathcal{V}_{th}$ in Eq. 13 is set to 0.3. We split a given time window to 5 bins, *i.e.*, n is set to 5 in Eq. 7. The impact of the hyperparameters α, $\mathcal{V}_{th}$ and n are discussed in the Supplementary Material.

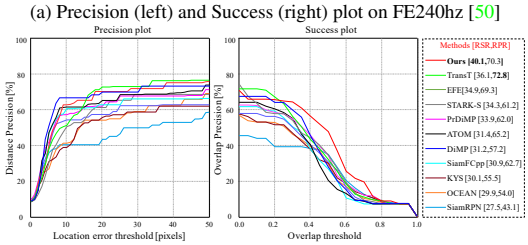
Evaluation Metrics Tracking performance is quantitatively measured by the following three metrics: success rate (SR), precision rate (PR), and overlap precision (OP_T). The SR focuses on the frame where the overlap between ground truth and predicted bounding box is larger than a threshold; OP_T matches SR but with a specific overlap threshold T. The PR counts the frame on which the center distance between ground truth and predicted bounding box is within a given threshold. We use the area under the curve as representative SR (RSR). Representative PR (RPR) is defined as a PR score associated with a 20-pixel threshold.

4.2. Evaluation

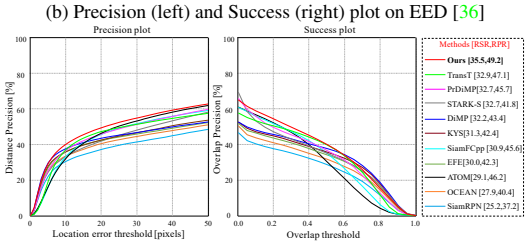
We evaluate the effectiveness of the proposed STNet from three perspectives: 1) overall tracking performance in terms of RPR and RSR, 2) tracking performance under different degraded conditions, and 3) tracking accuracy with respect to tracking speed. We compare our approach against ten existing state-of-the-art trackers on the FE240hz [50], EED [36], and VisEvent [16] datasets.



(a) Precision (left) and Success (right) plot on FE240hz [50]



(b) Precision (left) and Success (right) plot on EED [36]



(c) Precision (left) and Success (right) plot on VisEvent [16]

Figure 5. Precision (left) and Success (right) plot on the FE240hz, EED, and VisEvent datasets.

Overall Performance The overall tracking performance is reported in Figure 5 and Table 1, which indicate the proposed STNet offers state-of-the-art tracking performance on all three datasets. In particular, on the FE240hz dataset, STNet outperforms the runner-up by 1.8%, 3.9%, 2.8%, and 0.6% in RSR, $\text{OP}_{0.50}$, $\text{OP}_{0.75}$, and RPR, respectively. On the EED dataset, TransT [10] has a higher RPR than ours. However, our method outperforms the TransT in the other three metrics with a significant margin, 4.0% in RSP, 15.8% in $\text{OP}_{0.50}$, and 1.8% in $\text{OP}_{0.75}$. It should be noted that, compared to the FE240hz, EED is substantially less indicative due to the small set of annotations (1132K vs. 199 annotations). Finally, on the VisEvent dataset, the proposed STNet remains the best performer in RSR, $\text{OP}_0.50$, and RPR, outperforming the runner-up by 2.6%, 0.2%, and 2.1%, respectively. The analysis related to rigid and non-rigid objects is provided in the Supplementary Material.

Degraded Conditions Tackling real-world adverse conditions is one of the main motivations for using event-based cameras in object tracking tasks. We report the tracking performance of the top five methods on FE240hz in the following degraded conditions: (a) scenes with objects similar to the object being tracked; (b) severe camera motion; (c) scenes illuminated with a strobe light (*i.e.*, periodically turn on/off lights); and (d) high dynamic range (HDR) scenes. As shown in Figure 6, our approach fares best in all four

Dataset	Metrics	SiamRPN [28]	ATOM [13]	DiMP [4]	SiamFC++ [47]	OCEAN [51]	KYS [5]	PrDiMP [15]	STRAK-S [48]	TransT [10]	EFE [50]	STNet
FE240hz	RSR ↑	41.6	52.8	53.4	54.5	50.2	55.3	55.2	55.4	56.7	55.0	**58.5**
	$OP_{0.50}$ ↑	37.5	67.7	66.6	68.6	63.9	69.9	68.9	69.2	70.7	68.8	**74.6**
	$OP_{0.75}$ ↑	5.1	25.0	15.0	20.6	22.9	22.1	22.9	26.2	25.3	25.5	**28.1**
	RPR ↑	75.5	80.0	88.2	85.3	76.4	87.8	86.8	83.7	89.0	83.5	**89.6**
EED	RSR ↑	27.5	31.4	31.2	30.9	29.9	30.1	33.9	34.3	36.1	34.9	**40.1**
	$OP_{0.50}$ ↑	35.6	20.9	31.4	27.3	29.8	30.8	35.0	35.2	32.7	28.7	**48.5**
	$OP_{0.75}$ ↑	10.9	7.6	7.6	7.6	9.9	10.9	11.2	8.9	9.9	7.6	**11.7**
	RPR ↑	43.1	65.2	57.2	62.7	54.0	55.5	62.0	61.2	**72.8**	69.3	70.3
VisEvent	RSR ↑	25.2	29.1	32.2	30.9	27.9	31.1	32.7	32.7	32.9	30.0	**35.5**
	$OP_{0.50}$ ↑	28.5	30.0	37.4	34.8	31.5	35.5	36.3	34.8	39.5	33.9	**39.7**
	$OP_{0.75}$ ↑	15.4	7.1	**23.7**	11.2	18.7	22.9	17.5	21.4	18.0	21.1	20.4
	RPR ↑	37.2	46.2	43.4	45.6	40.4	42.4	45.7	41.8	47.1	42.3	**49.2**

Table 1. State-of-the-art comparison on the FE240hz [50], EED [36], and VisEvent [16] datasets in representative success rate (RSR), representative precision rate (RPR), and overlap precision (OP).

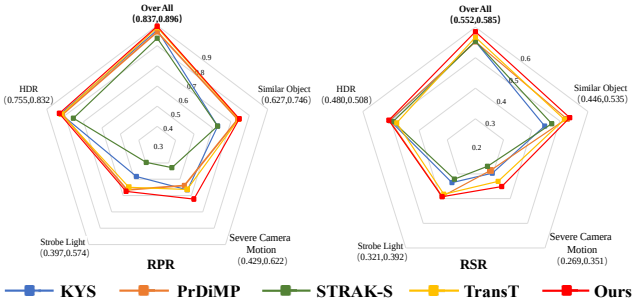


Figure 6. The top five methods tracking performance under four different adverse conditions provided by the FE240hz dataset [50]. (x, y) denotes a minimum value x and a maximum value y.

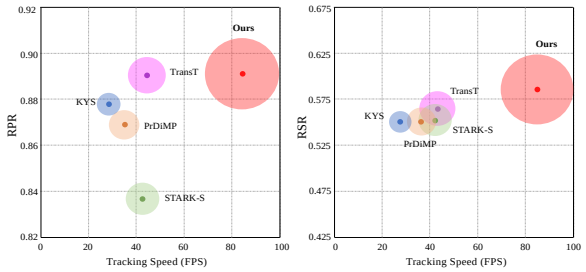


Figure 7. The RPR (left) and RSR (right) with respect to tracking speed of the top five methods on the FE240hz dataset [50].

degraded conditions. The two most challenging conditions are (b) and (c), and all competing methods struggle here. Under constant illumination, a static event-based camera is only sensitive to moving objects, making it an ideal sensor for tracking. However, when an event camera moves drastically (*i.e.*, condition (b)), almost all edges in a scene trigger events and make tracking challenging. In contrast, condition (c) offers no informative events when the light is off, which is challenging for all competing trackers. Figure 8 further qualitatively demonstrates the benefits of our method in these degraded conditions.

Tracking Speed We report the RPRs and RSRs with respect to the tracking speed of the top five methods on FE240hz in Figure 7 (larger area means better). The circle center's x and y coordinates indicate tracking speed and RPR/RSR, respectively. The results validate that our approach offers the best performance in both accuracy and speed. We note that the time for preprocessing event time is not included when estimating the tracking speeds of all trackers. We report the raw tracking speed evaluations in the Supplementary Material.

4.3. Ablation Study

To analyze our STNet, we investigate (a) the impact of spatial and temporal cues on single object tracking in the event domain; (b) the benefits of using SNN as a temporal feature extractor; and (c) the influence of each component in STNet. For each experiment, we re-train and test the modified models on the FE240hz [50] dataset. A complete list of ablation study experiments is reported in Table 2.

Impact of Spatial and Temporal Cues. We conduct the following experiments to demonstrate the effects of spatial and temporal cues on single object tracking with events: (A) Removing the transformer branch (TB) but keeping the SNN branch (SB) with a fixed spiking threshold; (B) The same settings as (A), except with dynamic spiking threshold; (C) Removing SB, but keeping TB. Comparing the original STNet (N) to (A), (B), or (C), we witness a significant performance degradation, validating that temporal and spatial cues are essential for tracking accuracy. The difference between (A) and (B) proves that the dynamic threshold is essential in extracting effective temporal cues.

	Networks	RSR ↑	$OP_{0.50}$ ↑	$OP_{0.75}$ ↑	RPR ↑
A	SFE w/ SB only w/o DT	55.2	69.5	20.9	86.2
B	SFE w/ SB only w/ DT	55.8	70.7	21.7	87.5
C	SFE w/ TB only	55.9	70.4	25.7	86.7
D	SFE SB → CNN-3	56.2	70.3	24.4	87.5
E	SFE SB → AlexNet	56.7	72.1	24.2	88.1
F	SFE SB → LSTM	57.1	71.8	23.7	89.2
G	w/o DTB	57.3	72.3	26.1	86.6
H	DTB w/o $\mathcal{K}$	57.9	73.2	27.8	89.0
I	DTB w/o $\mathcal{E}$	57.6	73.0	26.4	88.9
J	w/o TSFF	57.2	72.0	27.3	86.5
K	TSFF w/o TA	57.8	72.8	28.0	87.3
L	TSFF w/o CI	57.5	72.3	27.6	86.8
M	TSFF w/o SA	58.0	72.8	27.5	88.2
N	STNet	**58.5**	**74.6**	**28.1**	**89.6**

Table 2. Quantitative ablation comparisons: (a) leveraging SNN as temporal feature extractor improves single object tracking performance; (b) all components of STNet contribute to the overall performance. We denote SNN branch as 'SB'; Transformer branch as 'TB'. 'DTB' denotes dynamic threshold block;

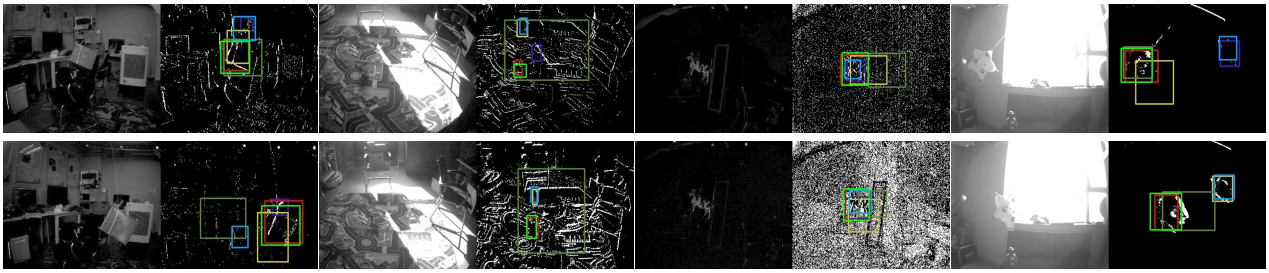


Figure 8. Qualitative comparison of STNet against SOTA trackers on the FE240hz [50] dataset. To better visualize the scene, we manually apply a gamma correction on the conventional frames (left). Note that only event frames (right) are used as input for all competing methods.

Benefits of SNNs. To provide further insights into the benefits of using SNN as a temporal feature extractor, we conduct three experiments by replacing the SNN branch of the SFE with (D) a 3-layer CNN, (E) an AlexNet [3], and (F) an LSTM [46] with one convolutional layer in each cell. We select these networks for fair comparisons as the SNN in the SB has three layers. Compared to the original STNet, we observe the accuracy degradation in all three experiments. As CNN and AlexNet [3] cannot grasp temporal cues, the performance degradation verifies the importance of temporal cues. In contrast, the LSTM [46] can extract temporal cues, resulting in better tracking performance but not as effective as the SNN. Notably, the spiking mechanism of an SNN neuron acts not only as temporal memory but also as a natural noise filter, which is beneficial to robust tracking.

Influence of STNet Components. We evaluate the influence of the SNN branch (SB) and transformer branch (TB) next. Here, we get insights into the dynamic threshold block (DTB), designed for dynamically assigning spiking thresholds. The effectiveness of the DTB is validated by the following experiments: (G) removing the entire DTB; (H) removing the representative value $\mathcal{K}$; and (I) removing the spatial entropy $\mathcal{E}$. Of the three experiments, (G) decreases the tracking performance the most. We also notice the performance degradation from (H) and (I) settings. These results reflect the effectiveness of the proposed DTB. More importantly, we show that leveraging spatial statistical cues for adjusting the threshold is an effective way to enforce spatial contextual information in an SNN.

Next, we investigate the impact of our TSFF module by removing it from the STNet. The corresponding experimental results are shown in the rows from J to M of Table 2. We see that the proposed TSFF module and its components contribute to STNet's tracking performance. We notice L as the worst performer in the last three settings, highlighting that the CI enhances the extracted features by bridging the temporal and spatial domains.

4.4. Limitations

Our method is not without limitations. When events are not available or too sparse, the effectiveness of our approach decreases, and Figure 9 shows such a case. However, other SOTA methods also suffer in these conditions. Designing methods that further leverage the SNN temporal memory may be a promising approach to tackling these failure modes.

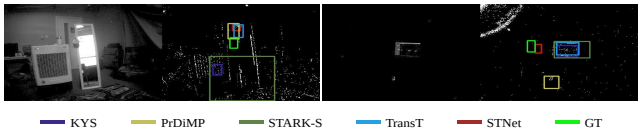


Figure 9. The performance of STNet falls off in scenes with sparse events, especially in the area of the object being tracked.

5. Conclusion

We present a spiking transformer network for single object tracking from event frames, STNet, that is capable of dynamically fusing temporal and spatial cues encoded in the events. The proposed network relies on several novel modules. We devise an SNN-based temporal feature extractor, which exploits statistical cues of spatial information to adjust the spiking threshold dynamically. We also introduce a novel temporal-spatial feature fusion module for dynamically fusing the features from the two domains, which relies on a novel cross-domain attention scheme. Extensive validation and ablation experiments verify the effectiveness and robustness of STNet in real-world scenarios. In addition, we experimentally confirm the effectiveness of SNNs as a temporal feature extractor and validate the benefit of adjusting spiking thresholds dynamically. The proposed STNet is the first in a line of work that explores learning dynamic spatio-temporal SNNs for event-based vision.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China under Grant 61972067/U21A20491/U1908214, National Key Research and Development Program of China (2021ZD0112400), and the Innovation Technology Funding of Dalian (2020JJ26GX036). Felix Heide was supported by an NSF CAREER Award (2047359), a Sony Young Faculty Award, and a Project X Innovation Award.

References

[1] Alex M Andrew. Spiking neuron models: Single neurons, populations, plasticity. *Kybernetes*, 2003. 2, 3

[2] Francisco Barranco, Cornelia Fermuller, and Eduardo Ros. Real-time clustering and multi-target tracking using event-based sensors. In *IROS*, 2018. 1, 3

[3] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, 2016. 3, 8

[4] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, 2019. 3, 7

[5] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Know your surroundings: Exploiting scene information for object tracking. In *ECCV*, 2020. 7

[6] Goutam Bhat, Felix Järemo Lawin, Martin Danelljan, Andreas Robinson, Michael Felsberg, Luc Van Gool, and Radu Timofte. Learning what to learn for video object segmentation. In *ECCV*, 2020. 3

[7] Luis A Camuñas-Mesa, Teresa Serrano-Gotarredona, Sio-Hoi Ieng, Ryad Benosman, and Bernabé Linares-Barranco. Event-driven stereo visual tracking algorithm to solve object occlusion. *IEEE Transactions on Neural Networks and Learning Systems*, 2017. 1

[8] Haosheng Chen, David Suter, Qiangqiang Wu, and Hanzi Wang. End-to-end learning of object motion estimation from retinal events for event-based object tracking. In *AAAI*, 2020. 1, 3

[9] Haosheng Chen, Qiangqiang Wu, Yanjie Liang, Xinbo Gao, and Hanzi Wang. Asynchronous tracking-by-detection on adaptive time surfaces for event-based object tracking. In *ACM MM*, 2019. 1, 3

[10] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *CVPR*, 2021. 3, 6, 7

[11] Zedu Chen, Bineng Zhong, Guorong Li, Shengping Zhang, and Rongrong Ji. Siamese box adaptive network for visual tracking. In *CVPR*, 2020. 3

[12] Leon N Cooper and Mark F Bear. The bcm theory of synapse modification at 30: interaction of theory with experiment. *Nature Reviews Neuroscience*, 2012. 3

[13] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *CVPR*, 2019. 7

[14] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, 2017. 3

[15] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *CVPR*, 2020. 7

[16] Wang et al. Visevent: Reliable object tracking via collaboration of frame and event flows. *arXiv*, 2021. 2, 6, 7

[17] Bertrand Fontaine, José Luis Peña, and Romain Brette. Spike-threshold adaptation predicted by membrane potential dynamics in vivo. *PLoS Computational Biology*, 2014. 3

[18] Guillermo Gallego, Tobi Delbruck, Garrick Michael Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jorg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1, 2

[19] Jin Gao, Weiming Hu, and Yan Lu. Recursive least-squares estimator-aided online learning for visual tracking. In *CVPR*, 2020. 3

[20] Daniel Gehrig, Michelle Rüegg, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Combining events and frames using recurrent asynchronous multimodal networks for monocular depth prediction. *IEEE Robotics and Automation Letters*, 2021. 2

[21] Wulfram Gerstner. Time structure of the activity in neural network models. *Physical Review E*, 1995. 2, 3

[22] Wulfram Gerstner and Werner M. Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. 2002. 2

[23] Samanwoy Ghosh-Dastidar and Hojjat Adeli. Spiking neural networks. *International Journal of Neural Systems*, 2009. 2, 3

[24] Arren Glover and Chiara Bartolozzi. Robust visual tracking with a freely-moving event camera. In *IROS*, 2017. 1, 3

[25] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. Siamcar: Siamese fully convolutional classification and regression for visual tracking. In *CVPR*, 2020. 3

[26] Tilman J Kispersky, Fernando R Fernandez, Michael N Economo, and John A White. Spike resonance properties in hippocampal o-lm cells are dependent on refractory dynamics. *Journal of Neuroscience*, 2012. 2, 3

[27] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E Shi, and Ryad B Benosman. Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016. 1

[28] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018. 3, 7

[29] Peixia Li, Boyu Chen, Wanli Ouyang, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Gradnet: Gradient-guided network for visual object tracking. In *ICCV*, 2019. 3

[30] Hongjie Liu, Diederik Paul Moeys, Gautham Das, Daniel Neil, Shih-Chii Liu, and Tobi Delbrück. Combined frame- and event-based detection and tracking. In *ISCAS*, 2016. 1

[31] Qianhui Liu, Dong Xing, Huajin Tang, De Ma, and Gang Pan. Event-based action recognition using motion information and spiking neural networks. In *IJCAI*, 2021. 3

[32] Yuhao Liu, Jiake Xie, Xiao Shi, Yu Qiao, Yujie Huang, Yong Tang, and Xin Yang. Tripartite information mining and integration for image matting. In *ICCV*, 2021. 5

[33] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *ICCV*, 2021. 2, 3, 4

[34] Ana I Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *CVPR*, 2018. 1

[35] Haiyang Mei, Bo Dong, Wen Dong, Pieter Peers, Xin Yang, Qiang Zhang, and Xiaopeng Wei. Depth-aware mirror segmentation. In *CVPR*, 2021. 5

[36] Anton Mitrokhin, Cornelia Fermüller, Chethan Parameshwara, and Yiannis Aloimonos. Event-based moving object detection and tracking. In *IROS*, 2018. 1, 2, 3, 6, 7

[37] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016. 3

[38] Ewa Piatkowska, Ahmed Nabil Belbachir, Stephan Schraml, and Margrit Gelautz. Spatiotemporal multiple persons tracking using dynamic vision sensor. In *CVPRW*, 2012. 1, 3

[39] Karine Pozo and Yukiko Goda. Unraveling mechanisms of homeostatic synaptic plasticity. *Neuron*, 2010. 3

[40] Yu Qiao, Yuhao Liu, Xin Yang, Dongsheng Zhou, Mingliang Xu, Qiang Zhang, and Xiaopeng Wei. Attention-guided hierarchical structure aggregation for image matting. In *CVPR*, 2020. 5

[41] Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. In *BMVC*, 2017. 1

[42] Sumit Bam Shrestha and Garrick Orchard. Slayer: Spike layer error reassignment in time. *ArXiv*, 2018. 2, 3

[43] Qian-Quan Sun. Experience-dependent intrinsic plasticity in interneurons of barrel cortex layer iv. *Journal of Neurophysiology*, 2009. 3

[44] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 3, 5

[45] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience*, 2018. 2, 3, 4

[46] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NeurIPS*, 2015. 8

[47] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *AAAI*, 2020. 4, 5, 6, 7

[48] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. *ICCV*, 2021. 3, 7

[49] Jiqing Zhang, Chengjiang Long, Yuxin Wang, Haiyin Piao, Haiyang Mei, Xin Yang, and Baocai Yin. A two-stage attentive network for single image super-resolution. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022. 5

[50] Jiqing Zhang, Xin Yang, Yingkai Fu, Xiaopeng Wei, Baocai Yin, and Bo Dong. Object tracking by jointly exploiting frame and event domain. In *ICCV*, 2021. 1, 2, 3, 6, 7, 8

[51] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *ECCV*, 2020. 3, 7