Distractor-Aware Event-Based Tracking

Yingkai Fu, Meng Li, Wenxi Liu, Yuanchen Wang, Jiqing Zhang, Baocai Yin, *Member, IEEE*, Xiaopeng Wei, and Xin Yang

Abstract— **Event cameras, or dynamic vision sensors, have recently achieved success from fundamental vision tasks to high-level vision researches. Due to its ability to asynchronously capture light intensity changes, event camera has an inherent advantage to capture moving objects in challenging scenarios including objects under low light, high dynamic range, or fast moving objects. Thus event camera are natural for visual object tracking. However, the current event-based trackers derived from RGB trackers simply modify the input images to event frames and still follow conventional tracking pipeline that mainly focus on object texture for target distinction. As a result, the trackers may not be robust dealing with challenging scenarios such as moving cameras and cluttered foreground. In this paper, we propose a distractor-aware event-based tracker that introduces transformer modules into Siamese network architecture (named DANet). Specifically, our model is mainly composed of a motion-aware network and a target-aware network, which simultaneously exploits both motion cues and object contours from event data, so as to discover motion objects and identify the target object by removing dynamic distractors. Our DANet can be trained in an end-to-end manner without any post-processing and can run at over 80 FPS on a single V100. We conduct comprehensive experiments on two large event tracking datasets to validate the proposed model. We demonstrate that our tracker has superior performance against the state-of-the-art trackers in terms of both accuracy and efficiency.**

Index Terms— **Event camera, visual object tracking, vision transformer, deep neural network.**

I. INTRODUCTION

VISUAL object tracking is an important topic in computer vision filed, where the target location is known in the first frame of the video and the algorithm has to identify the target for the rest frames. However, there still remains challenging for the current tracking algorithms and sensors to achieve better

Yingkai Fu, Yuanchen Wang, Jiqing Zhang, Xiaopeng Wei, and Xin Yang are with the Key Laboratory of Social Computing and Cognitive Intelligence, Ministry of Education, Dalian University of Technology, Dalian 116024, China (e-mail: yingkai.fu@outlook.com; wangyc0604@mail.dlut.edu.cn; jqz@mail.dlut.edu.cn; xpwei@dlut.edu.cn; xinyang@dlut.edu.cn).

Meng Li is with HiSilicon (Shanghai) Technologies Company Ltd., Shanghai 201799, China (e-mail: limeng8@hisilicon.com).

Wenxi Liu is with the College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China (e-mail: wenxi.liu@hotmail.com).

Baocai Yin is with the Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (e-mail: ybc@bjut.edu.cn).

accuracy in the environment where the illumination is unstable and the target moves at high speed. On the contrary, the event camera is a novel bionic sensor designed to imitate the imaging mechanism of the human retina. Comparing to the traditional vision sensors like commodity RGB cameras, it has high temporal resolution, very high dynamic range, low power consumption and high bandwidth [1]. In recent years, event cameras have been employed in more and more areas including object classification [2], object detection [3], robotic collision avoidance [4], [5], video reconstruction [6], [7], [8], [9], SLAM [10], [11], [12], [13], [14], etc.

However, despite the obvious advantages of event cameras, designing a visual object tracking algorithm based on event cameras is still challenging. Event cameras can capture changes in light intensity. Typically, they generate more events along the edges of moving objects, where there are larger variations in light intensity, while fewer events occur within the object boundaries where the light intensity changes less. As a result, event cameras cannot extract texture information from the internal of the target effectively, thus most RGB-based trackers that identify and track objects by rich object texture information are not suitable for event-based tracking. Proposing new pipelines for event-based tracking is crucial. References [15] and [16] distinguish between background and moving targets by clustering algorithms, however, the introduced handcrafted strategies are not generalizable across scenarios. Recently, a line of works [17], [18], [19] relies on deep learning to improve the capability of feature extraction, thereby enhancing the discrimination of targets. However, these methods do not explicitly model the relationship between targets and motion. Therefore, when the scene contains distractors such as similar co-moving objects or background noises, the approach is not effective in locating the target.

In this work, we consider both the target distinction and movement information provided by event-based camera for robust tracking, where the target distinction provided by edge information aims to identify the target, and the movement information helps continuously track the object [20]. As shown in Figure 1, we propose a novel event-based object tracking framework (named DANet). In particular, we mainly divide the tracking process in two stages, i.e., ***discovery*** and ***identification***, in which ***discovery*** aims to find out the potential dynamic objects from current event frame, while the goal of ***identification*** is to identify the target object with templates. Accordingly, our DANet is mainly composed of two subnetworks: motion-aware network and target-aware network. In specific, the motion-aware network observes the current
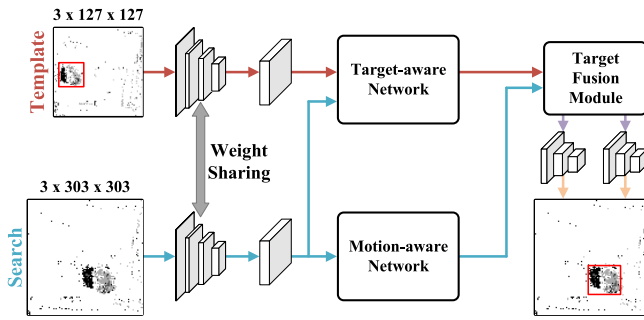
Fig. 1.   The overview of the proposed event-based tracking pipeline. The target-aware network identifies and extracts the target embedding by matching the object edge, the motion-aware network extract moving object features. The target motion fusion module fuses the target embedding and moving objects to locate the moving target. Both the motion and edge information of the event camera can be utilized for precise tracking with the proposed network.

search frame to distinguish the dynamic objects in the scene. The scene dynamics are extracted via convolutional layers and then passed into transformer-based modules in which the self-correlation of scene features can be computed to discover the significant contrast regions between foreground objects and background. On the other hand, the target-aware network is fed with both the template frame and search frame, and identifies the target object with the help of the template frame. Concretely, the response maps for the template frame and the search frame are extracted through the correlation, then the initial response maps are refined through the transformer-based encoder-decoder module to establish the correlation between target object events and contextual events, resulting in the target embeddings of the target object.Finally, the motion feature from the motion aware network is guided by the output target embedding from the target aware network to generate a confidence map for localizing the target objects. Comprehensive experiments are conducted on public event-based tracking datasets include FE240 dataset [18] and VisEvent dataset [21]. Experimental results show our DANet achieves the state-of-the-art performance in accuracy and efficiency, compared with prior approaches. Overall, the main contributions include:

- We propose a novel event-based tracking model that exploits motion cues and object contours to discover and identify target objects, which facilitates awareness and discrimination in challenging tracking scenarios.
- Our proposed tracker consists of two sub-networks: motion-aware network and target-aware network. Motion-aware network employs transformer-based module to extract scene dynamics, while target-aware network correlates the current frame and search frame to remove distractors.
- The comprehensive experiments on FE240 and VisEvent datasets show that our proposed framework can achieve state-of-the-art performance compared to existing trackers.

The rest of this paper is organized as follows. Section II introduce the fundamentals and related works of vision transformer and the visual object tracking task. In Section III, we explain the details of the proposed DANet. The experimental results, ablations and analysis are provided in Section IV. We conclude the paper in Section V.

II. RELATED WORK

A. Vision Transformer

Recently, the vision transformer becomes popular as its ability to capture long range information. It has been applied for many computer vision tasks, like image super-resolution [22], [23], [24], object detection [25], [26], [27], [28] and visual object tracking [29], [30], [31]. Transformer are designed to replace the recurrence and convolutions entirely with muti-head self-attention mechanism to capture global information and reallocate the features [29]. The attention mechanism tries to enhance the regions while fading out the non-relevant information by learning relationships between elements of a sequence while keeping their entire composition intact with query, key and value [32]. In this work, we utilize the transformer architecture to decode target embedding and encode motion cues so as to exploit the contrast difference between dynamic target and background from global perspective for distractor-aware event-based tracking.

B. Visual Object Tracking

Visual object tracking is a central topic of discussion in computer vision, which aims at predicting the target position given the initial state (*i.e.*. the bounding box of the target). The task remains challenging as the appearance of tracking target changes due to light variation, and similar moving target [33], [34], [35], [36]. The mainstream deeplearning-based RGB tracking approaches can be divided into three parts, the discriminative correlation filter (DCF) based tracker that leverage a correlation-based network to compute the similarity between the template and the search region for target tracking [37], [38], [39], [40], [41], [42], [43], the Siamese-like tracker, which learn a model predictor and use it for judge target [44], [45], [46] and transformer-based tracker [29], [30], [47]. We shows the differences of these pipelines in Figure 2(A, B, C). These tracking pipelines work well in the RGB domain as the RGB frame can supply enough texture information for feature matching and object discrimination, however, for scenes like low light, HDR and fast motion, these pipelines can not work efficient as the visual cues are insufficient. To deal with the above nature defects of RGB camera, there are increasing works use event camera for visual tracking to settle scenarios with challenging light variation. In traditional-based methods, Mondal et al. [48] presented an unsupervised graph spectral clustering technique for moving object location in event-based data. Barranco et al. [15] proposed a real-time mean-shift clustering algorithm for event-based multi-object tracking. In deep learning-based methods, Chen et al. [17] proposed a novel event representation to encode the spatio-temporal information and designed a 5-DoF object-level motion model to regress object motions. In multi-domain event-based tracking tasks, Zhang et al. [18], [49] proposed attention scheme and multi-modality alignment to fuse the frame domain and event domain for single object tracking, Zhu et al. [50] proposed a mask modeling strategy on pre-trained vision transformer to enforce the interaction from RGB and Event modalities, Wang et al. [51] proposed a unified trmer backbone network for color-event tracking. However,
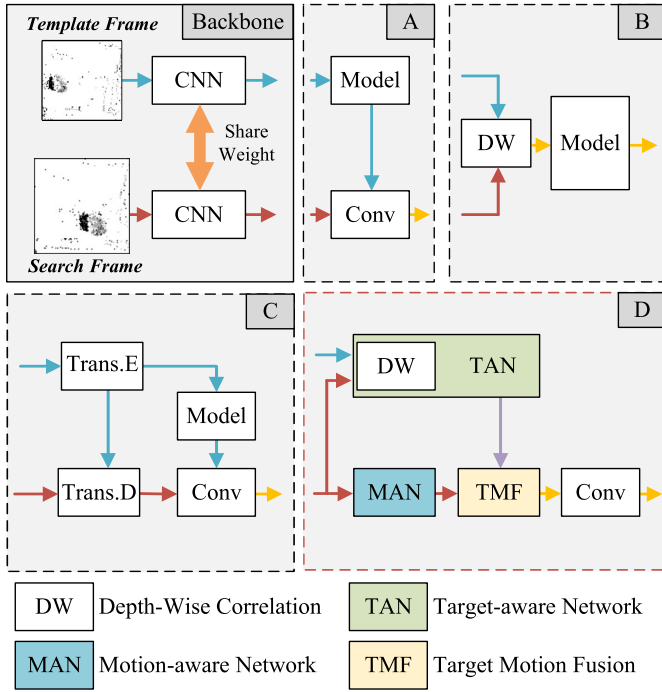
Fig. 2. A simplified demonstration of mainly trackers, which can be divided into Siamese-like tracker (A), DCF tracker (B) and transformer tracker (C) [30]. In contrast to the existing trackers. our proposed tracker (D) is designed to make use of both the appearance cue and motion cue from the event camera for robuster event-camera based object tracking.

these event tracker are designed to leverage the edge cues, therefore, when the camera is moving, these trackers failed to distinguish target from background. Actually, an event-based camera does not generate semantic information, it offers rich object motion information as moving objects can typically generate more events due to the large contrast changes at the object edge [1], [18]. In this paper, we exploit the motion cues for robust tracking.

III. OUR PROPOSED METHOD

A. Framework Overview

As illustrated in Figure 1, to facilitate the **discovery** and **identification** ability of the proposed pipeline, our framework is mainly composed of feature extraction network, target-aware network, motion-aware network, target fusion module, as well as object localization regressor. In particular, the feature extraction module first extracts the features for the template and search frames. Then, the features of the search frame are fed into the motion-aware network that extracts scene dynamics, while those of template and search frames are both sent into the target-aware network so as to remove dynamic distractors and focus on the target object. Finally, their outcomes are fused in the target fusion module and employed to regress the target location. We will describe the details of each component in the remainder of this section. The details of the target-aware network, the motion-aware network and target fusion module are visualized in Figure 3.

B. Event Representation

Different from conventional frame-based cameras, event-based cameras produce asynchronous data streams at high speed, when any pixel detects a log scale intensity change that exceeds a threshold C, as:

$$L(x, y, t) - L(x, y, t - \Delta t) \geq pC, \qquad (1)$$

where L denotes log scale light intensity, C is the contrast threshold determined by the hardware, and $p \in \{-1, +1\}$ is the event polarity depending on the change in brightness. Mathematically, a set of events can be described as:

$$E = \{e_k\}_{k=1}^{N} = \left\{ \left[x_k, y_k, t_k, p_k \right] \right\}_{k=1}^{N}, \qquad (2)$$

where k is the index of each event, x_k and y_k represent the pixel location, t_k denotes the timestamps of the event.

Yet, the asynchronous events cannot be processed directly due to their sparse nature, and thus algorithms designed for conventional frames are not applicable. To address this issue, for each pixel, we propose to sum up the polarity of sparse event data within a fixed time interval to obtain event frames, The event stream and the stacked event frame are illustrate in Figure 4, the stacking process can be formulated as:

$$F(x, y, \Delta t) = \sum_{k=i}^{N} p_k(x, y), \qquad (3)$$

There are several benefits for aggregating event data in this way. First, the original sparse event stream can be enhanced, while the event frames can still reflect object contours as the events caused by little contrast change will be filtered out. Second, the aggregated offset values in event frames can reflect the extent of object motion, which can be used to better identify different moving objects. Besides, the event aggregation can be efficiently computed in parallel and it is easy to integrate it into existing event camera development kits without compromising real-time data processing and tracking capabilities.

C. Event Feature Extraction

Based on Siamese network architecture, our event-based feature extraction module is fed with the template frame, $z \in \mathbb{R}^{H_z \times W_z}$, and the search template, $x \in \mathbb{R}^{H_x \times W_x}$. Both frames are passed into a shared network based on ResNet-18 [52] for extracting event features. Thus, we can obtain the event features of z, i.e., $f(z) \in \mathbb{R}^{C \times \frac{H_z}{s} \times \frac{W_z}{s}}$, and those of x, $f(x) \in \mathbb{R}^{C \times \frac{H_x}{s} \times \frac{W_x}{s}}$, where s is downsampling ratio. These features are then input to the proposed Target-aware Network and Motion-aware Network for extracting the similar target and moving objects.

D. Target-Aware Network (TAN)

The detail of the target-aware network is illustrated in Figure 3(B), the TAN aims to match the given tracking target at the template frame with the search frame. To be specific, firstly, the features of the template frame $f(z)$ and the search frame $f(x)$ are separately passed to two convolution blocks with 3×3 kernel size. Then, the template feature embeddings are employed as a kernel to compute convolution on the feature embeddings of the search frame to produce an initial target response map R, which implies the confidence of object
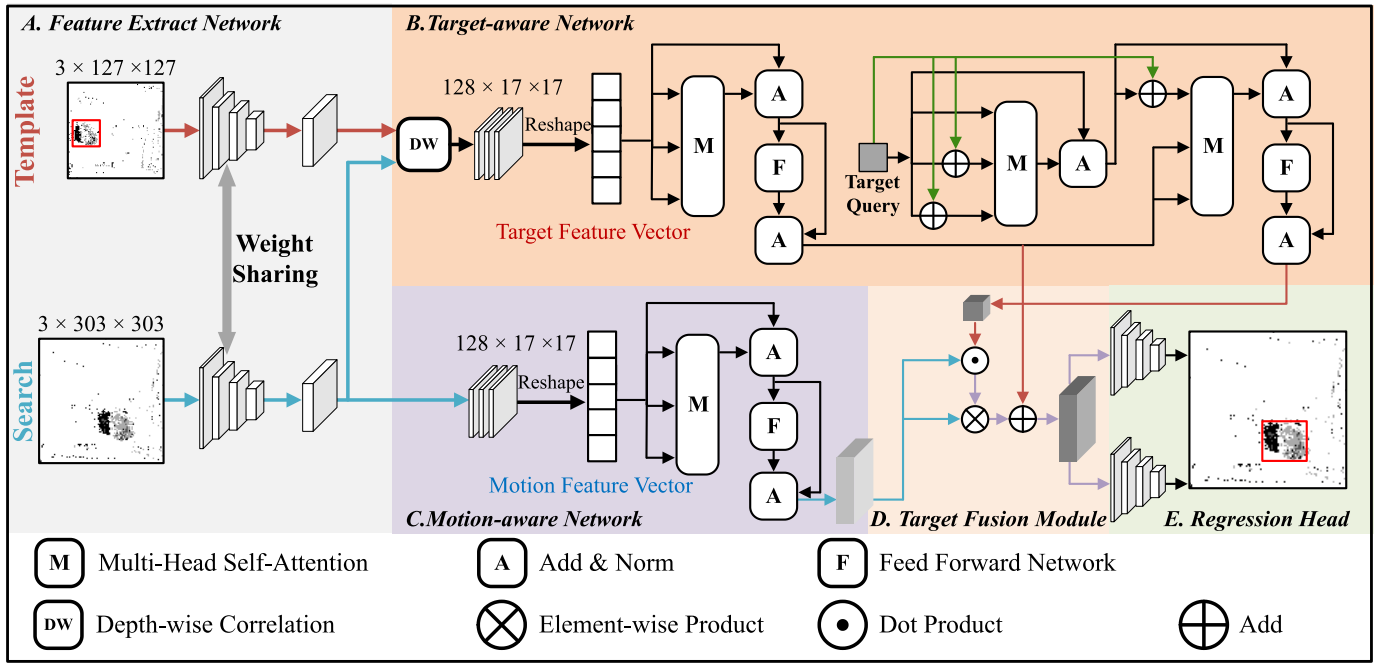
Fig. 3. The details of five main parts proposed in our tracker, namely feature extract network, target-aware network, motion-aware network,target fusion module and regression head. The search feature extract from search frame come to both target-aware network and motion-aware network to provide target features and movement features.
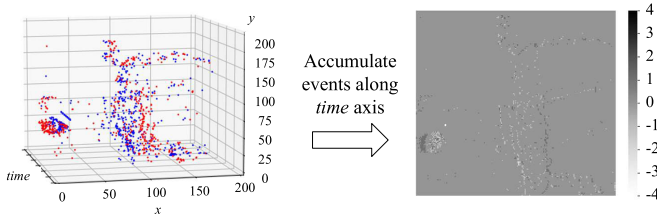

Fig. 4. The event steam is accumulated along t axis to generate an event frame, the moving part can be emphasized and some background noises are filtered out.

localization. Formally, we have:

$$R = \psi(z, x) = \phi(f(z)) \star \varphi(f(x)), \qquad (4)$$

where $f(\cdot)$ refers to feature extraction network. $\phi(\cdot)$ and $\varphi(\cdot)$ are two convolutional layers. $\star$ represents the depth-wise cross-correlation operator, which enables to imply the coarse correlation between the contour of target object and the events within search region. To refine the initial target response map while removing distractors, we present transformer-based modules to strengthen the event information from a global perspective. Firstly, we review multi-head attention mechanism, as:

$$MultiHead(Q, K, V) = Concat(H_1, , H_n)W^o \qquad (5)$$

$$H_i = Attention(QW_i^Q, KW_i^K, VW_i^V), \qquad (6)$$

$$Attention(Q, K, V) = \text{softmax}(QK^\top/\sqrt{d_k})V, \qquad (7)$$

where $W_i^Q \in \mathbb{R}^{d_m \times d_k}$, $W_i^K \in \mathbb{R}^{d_m \times d_k}$, $W_i^V \in \mathbb{R}^{d_m \times d_v}$, and $W^O \in \mathbb{R}^{n_h d_m \times d_m}$ denote model parameters.

In TAN, we use Transformer to enable target embedding. To do so, the initial target response map is first passed through a bottleneck layer to squeeze the features by reducing the

channel number from C to d, and then flatten the feature maps along the spatial dimension to form a feature vector. Next, the correlation R is fed to the encoder blocks to utilize the contextual information. The encoder block consists of multi-head self-attention modules and feed forward network with two linear transformation with a ReLU in between to enhance the fitting ability of the model. The output R' of an encoder block can be described as:

$$R' = \mathcal{F}_e + FFN(\mathcal{F}_e), \qquad (8)$$

$$\mathcal{F}_e = R + MultiHead(R, R, R), \qquad (9)$$

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2, \qquad (10)$$

where W and b are weight matrix and basis vectors, respectively. We further exploit a decoder module to obtain the embeddings of the target object, the inputs for a decoder module is a learnable target object query and the output from encoder. The decoder module can be expressed as below:

$$T = \mathcal{F}_d + FFN(\mathcal{F}_d), \qquad (11)$$

$$\mathcal{F}_d = A + MultiHead(A + TQ, R', R'), \qquad (12)$$

$$A = TQ + MultiHead(TQ + TQ, TQ + TQ, TQ), \qquad (13)$$

where TQ is a target query. We only input one single query into the decoder to predict one bounding box of the target object as in [47].

E. Motion-Aware Network (MAN)

As introduced in Section III-B, by accumulating the events within a short time interval into an event frame and adding up the polarity for each pixel, the ability to capture moving objects can be preserved. Therefore, event frames can be used for discovering all the moving objects within the scene, since

moving objects generally trigger more events compared to the background.

To accomplish this, the extracted features of the search frame are firstly passed to a 3×3 convolutional layer with stride 2 to compress and reshape the features to M, it is further sent into the motion-aware network (see Figure 3(C)). The structure of the motion-aware network consists of the transformer encoder module that is identical to the target-aware network. The motion-aware network works as:

$$M' = \mathcal{M}_e + FFN(\mathcal{M}_e), \tag{14}$$

$$\mathcal{M}_e = M + MultiHead(M, M, M). \tag{15}$$

F. Target Fusion Module (TFM)

In the final stage, we propose to fuse the target embeddings T and the motion feature maps M' to generate the refined heat map for regressing moving target. Specifically, as shown in Figure 3(D), we take target embedding from the output of target-aware network, then compute the similarity between the target features and the output motion features from the motion-aware network. Next, we apply the similarity scores to the motion features with an element-wisely multiply to enhance the regions for the target objects and weaken other moving objects. To effectively utilize the output of the encoder in TAN, we introduce a residual connection after multiplication to fuse two outcomes, the output feature maps are then used to regress object boundaries.

G. Regression Head

The regression head aims to locate the target with the enhanced feature maps. The enhanced feature is passed through two fully convolution networks with ReLU activation function to formulate the features for predicting object center and the scale. Finally, we apply the soft-argmax function to extract coordinates via calculating a differentiable expectation [53]. Formally, we have:

$$c_x(h) = \sum_{i=1}^{W} \sum_{j=1}^{H} \mathrm{softmax}\left(h_{i,j}\right) w_{i,j,x}, \tag{16}$$

$$c_y(h) = \sum_{i=1}^{W} \sum_{j=1}^{H} \mathrm{softmax}\left(h_{i,j}\right) w_{i,j,y}, \tag{17}$$

$$w_{i,j,x} = \frac{i}{W} \times s, \, w_{i,j,y} = \frac{j}{H} \times s, \tag{18}$$

where $h_{i,j}$ denotes the output value at location (i, j) on feature maps h, s denotes the total stride that down sample from the original images size to the size of feature maps.

H. Loss Function

DANet does not require any postprocessing, therefore, it can be trained in an end-to-end fashion. We use the combination of L_1 Loss and the generalized IoU loss [54] as in [25] to regulate the training to predict target location. The loss function is formulated as:

$$L = \lambda_{\mathrm{iou}} L_{\mathrm{iou}}\left(b_i, \hat{b}_i\right) + \lambda_{L_1} L_1\left(b_i, \hat{b}_i\right), \tag{19}$$

where b_i and $\hat{b}_i$ represent the ground-truth and the predicted box. λ_{iou} and λ_{L_1} are balancing weights.

IV. EXPERIMENTAL RESULTS

A. Datasets and Implementation Details

a) Datasets: We use three event camera tracking datasets to validate the proposed tracker, namely FE240 dataset [18], VisEvent dataset [21] and EED dataset [55].

FE240 dataset is a large indoor event camera tracking dataset that contains various degraded scenarios, such as high dynamic range, low light, motion blur, and fast motion. It features 82 long sequences. All of them are captured by DAVIS346 event camera, which simultaneously capture events and calibrated frames. The ground-truth bounding boxes of the targets are annotated by a motion capture system at 240Hz. We use 57 sequences to train and the rest 25 sequences for testing.

VisEvent dataset is a large-scale neuromorphic tracking dataset collected from real world by a DAVIS346 color event camera. The VisEvent dataset includes challenging scenarios in various condition, *e.g.*, low illumination, high speed, clutter background, and slow motion. The sequences from this dataset contain RGB frames and events at the resolution of 346×260. The annotation for each frame is fulfilled by a professional label company at about 25 FPS. We exclude 97 training sequences and 23 test sequences from the original dataset due to wrong annotations or missing events, so there are 403 training sequences and 297 testing sequences used in our experiments.

EED dataset is a validate only event-based tracking dataset collected by a DAVIS event camera under extreme condition such as strobe light, occlusion and cluster background. As the annotation for EED dataset is not precise, we manually label the dataset and generate event frame from the provided event files.

b) Metrics: We follow the experimental protocols of visual tracking as one-pass evaluation (OPE) [56] and report the following metrics to show the performance of the proposed method, *i.e.*, Area Under Curve (AUC), Overlap Precision (OP), Precision, and Normalized Precision, respectively. We also show the success plot and normalized precision plot for a comprehensive comparison of different thresholds.

Success plot shows the percentage of frames with the overlapping rate $S \geq t_0$ throughout all the thresholds $t_0 \in [0, 1]$. The overlapping rate is defined as:

$$S(a, b) = \frac{\mathrm{Area}(a) \cap \mathrm{Area}(b)}{\mathrm{Area}(a) \cup \mathrm{Area}(b)}, \tag{20}$$

where $\mathrm{Area}(a)$ denotes the bounding box predicted by the tracker, and the $\mathrm{Area}(b)$ means the bounding box given by the ground-truth. The AUC of each success plot serves as the second measure to rank the trackers. The success curve offers a continuous measurement of tracking results ranging from robustness (lower overlap rate but more tracked frames) to accuracy (higher overlap rate) [57]. The overlap precision metric (OP_T) is defined as the percentage of frames where bounding box IoU overlap larger than a threshold T.
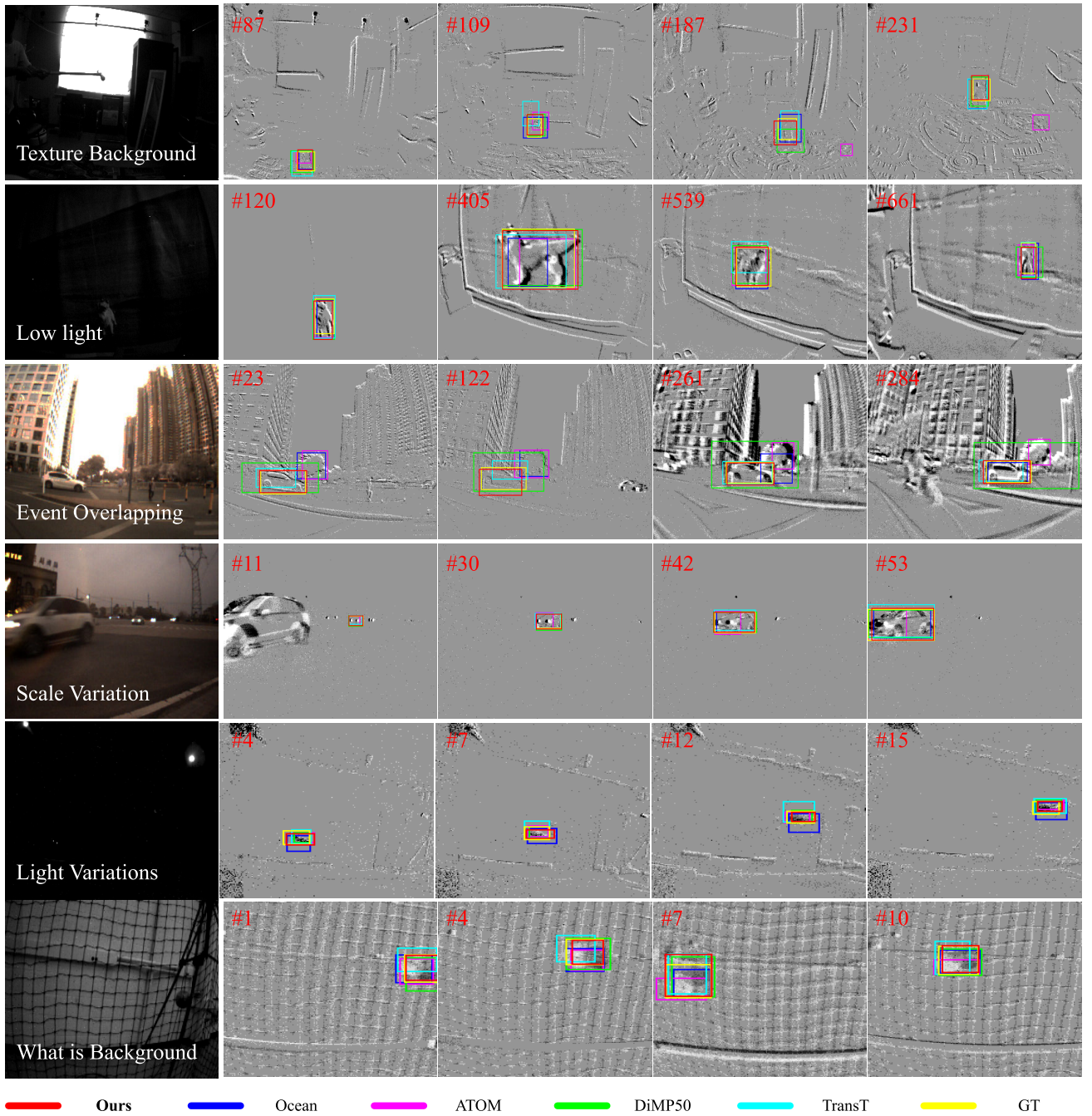
Fig. 5. Visual comparison of our proposed tacker against other state-of-the-art trackers on FE240 [18] dataset (The first two rows), VisEvent [21] dataset (row 3-4) and EED dataset [55] (row 5-6). The images in first column are only used for visualization.

Precision plot measures the distance between the center of predicted bounding box and that of the ground-truth bounding box. However, the target size and image resolution have large discrepancies for different frames and videos, which will heavily influence the precision metric. Therefore, we only use normalized precision plot for comprehensive comparison for each threshold. The precision rate with the threshold 20 and the normalized precision rate with the threshold 0.2 are used to rank the trackers.

c) Implementation details: The inputs for our tracker are a pair of frames, *i.e.*, the template frame and search frame which are cropped and resized to 127×127 and 303×303, respectively. The feature extractor backbone ResNet18 [52]

is pretrained on ImageNet [58]. The following convolutional blocks consist of three stacked Conv-BN-ReLU layers to reduce the feature size and produce the feature maps of $128 \times 17 \times 17$. The head number of multi-head attention is set to 4 with the width 128. In addition, TAN consists three encoders and three decoders, while MAN consists of two encoder blocks. TAN takes an additional target query as input. The feed-forward network in both TAN and MAN has a hidden dimension of 2048 with a dropout ratio of 0.1. The FCN layers in the object localization regressor are five stacked Conv-BN-ReLU layers with the output channel as 128, 64, 32, 16 and 1, respectively, which produce two feature maps that are separately used to produce the center coordinates and the

TABLE I

STATE-OF-THE-ART COMPARISON ON FE240 DATASET AND VISEVENT DATASET IN TERMS OF AUC, PRECISION AND NORMALIZED PRECISION. THE FIRST, SECOND, AND THIRD BEST RESULTS ARE MARKED IN RED, GREEN, AND BLUE, RESPECTIVELY. OUR PROPOSED METHOD ACHIEVES THE BEST PERFORMANCE ON THESE TWO DATASET UNDER FIVE STANDARD COMPARING METRICS

Dataset	Metrics	STNet [19]	OCEAN [62]	ATOM [44]	TransT [29]	Stark_S [47]	SiamFC++ [61]	DiMP50 [45]	PrDiMP50 [46]	DANet
		CVPR'22	ECCV'20	CVPR'19	CVPR'21	ICCV'21	AAAI'20	ICCV'19	CVPR'20	Ours
FE240 [18]	Success(AUC)(%) ↑	31.51	47.28	40.07	53.46	51.79	45.23	54.49	53.43	56.86
	OP50.(%) ↑	36.07	55.93	50.08	64.35	61.68	56.17	68.79	65.98	69.94
	OP75.(%) ↑	6.92	11.99	16.80	20.06	19.88	13.64	17.63	20.13	25.03
	Precision(%) ↑	52.46	80.96	61.85	86.13	83.9	74.23	87.55	85.14	89.19
	Norm. Prec.(%) ↑	31.78	47.47	41.93	52.11	49.43	47.29	55.88	52.28	58.37
VisEvent [21]	Success(AUC)(%) ↑		34.18	32.82	37.30	34.97	37.30	38.03	38.35	39.76
	OP50.(%) ↑		39.25	37.01	44.63	39.86	43.81	43.80	44.84	45.62
	OP75.(%) ↑		19.12	20.96	21.37	23.61	23.71	25.02	27.18	30.27
	Precision(%) ↑		51.39	46.48	54.82	48.19	53.08	54.08	53.93	54.54
	Norm Prec.(%) ↑		50.24	45.2	51.15	46.80	52.36	51.58	52.66	54.07
EED [55]	Success(AUC)(%) ↑	35.50	53.45	48.36	48.98	42.63	42.81	53.60	48.62	55.69
	OP50.(%) ↑	39.37	54.56	48.21	53.4	42.72	55.73	65.64	46.76	65.94
	OP75.(%) ↑	10.79	12.36	12.97	12.09	8.41	10.79	19.94	10.44	23.43
	Precision(%) ↑	62.70	100	85.58	88.70	75.47	69.47	86.78	91.22	86.78
	Norm Prec.(%) ↑	31.25	41.29	58.50	45.46	36.59	38.83	58.53	57.03	50.56

shape of target object. We set λ_{iou} to 1 and λ_{L_1} to 5 in the loss function.

d) Training and network inference: Our tracker is implemented using Python 3.8 and PyTorch 1.9.1 [59]. The experiments are conducted on a server with Intel(R) Xeon(R) Gold 6240 CPU @ 2.60GHz, and an NVIDIA V100 32GB GPU with CUDA 11.1. At the training stage, it takes a warm up [60] epoch with the learning rate linearly increased from 10^{-5} to 8×10^{-2}, and then uses a cosine annealing learning rate schedule for the rest 19 epochs. We choose stochastic gradient descent as the optimizer. The batch size is set to 128. For each epoch, 3×10^5 images pairs are sampled randomly. During inference, given the first event frame and the corresponding bounding box, the network initializes the template branch and calculates the template features. The input search frames are cropped around the latest predicted location. Then, the Siamese feature extractor extracts features for the search event frame. After that, the search features and the template features are sent to the succeeding modules to predict a bounding boxes, which represent the normalized location relative to the previous prediction location. Finally, a simple transformation is applied to restore the box scale that fit the original image size.

B. Comparison With SOTA Trackers

To verify the efficiency and precision of our proposed DANet, we choose 8 SOTA trackers to be compared with, namely SiamFC++ [61], ATOM [44], DiMP50 [45], PrDiMP50 [46], STARKS [47], TransT [29], Ocean [62] and STNet [19]. For a fair comparison, we use their publicly available codes and then retrained these trackers with their default parameters. To validate the performance of each tracker, we report AUC, OP50, OP75, Precision, and Normalized Precision to demonstrate the performance of the tracker. The threshold of Precision and Normalized Precision is set to 20 pixels and 0.2, respectively.

a) Comparison on FE240 dataset: As shown in Table I, we can see that our proposed offline tracker surpasses other state-of-the-art trackers on all five evaluation metrics.
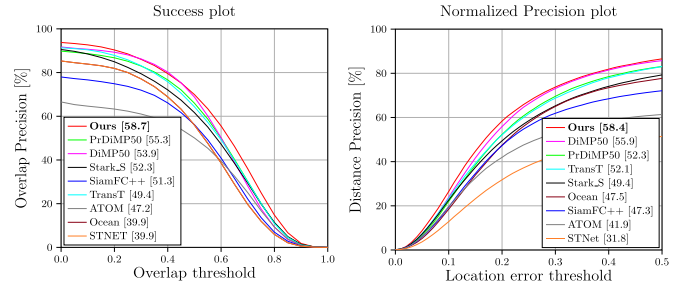

Fig. 6. Success plot and normalized precision plot on FE240 dataset [18]. In the legend, the area-under-curve (AUC) and normalized precision (P_{Norm}) are reported in the left and right figures, respectively. Our proposed method achieve the best performance in both two metric.

In particular, our tracker obtains a performance with 56.86%, 89.19% and 58.37% in terms of AUC, Precision, and Normalized Precision, respectively. Note that DANet excels DiMP50 [46] and PrDiMP50 [46] by a large margin, which is the state-of-the-art online tracker with ResNet-50 [52] as backbone. We also report the success plot and normalized precision plot in Figure 6. Additionally, we analyze the results on camera motion sequences and multiple object motion sequences. The results are displayed in Figure10. When the camera is moving or multiple objects are moving together, the events from the background or other objects may severely disturb the competitor. However, our tracker still achieves 48.69% and 61.94% for these two kinds of scenarios in terms of AUC. For scenes where illuminations are challenging, *e.g.*, HDR, low light and strobe light, our tracker still achieve good results. The visualization of the severe motion and strobe light scenes can be observed on the first and third row in Figure 11. Although the events from the target are overlapped with background events, our tracker keeps focusing on the moving target. More visual examples are shown in Figure 5 row 1-2, which indicates the superior performance of the proposed tracker in challenging scenes.

b) Comparison on visevent dataset: Since the VisEvent dataset [21] has many sequences that target objects are similar, static, fully occluded, or completely leave the video sequences, all the trackers suffer from performance degradation. However,
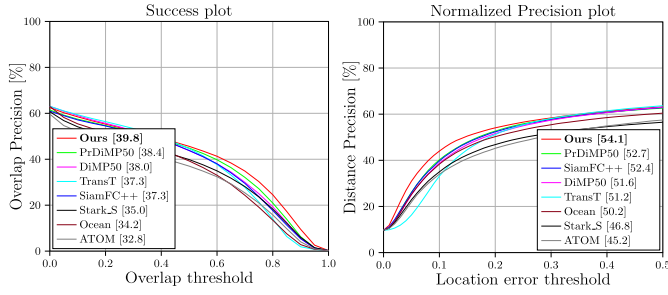
Fig. 7. Success plot and normalized precision plot on VisEvent dataset [21]. Our method has competitive performance compared to other trackers.
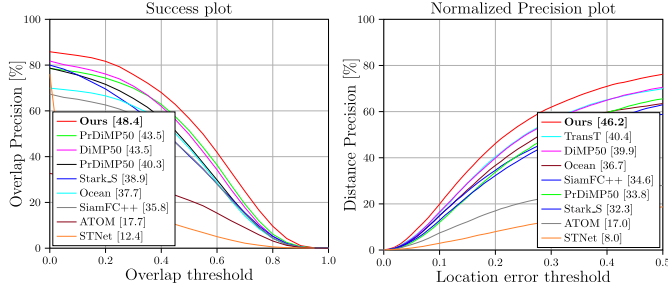

Fig. 8. Success plot and normalized precision plot of camera motion sequences on FE240 dataset [18], which demonstrates the proposed tracker has superior tracking performance compared to other algorithms in camera motion scenarios.
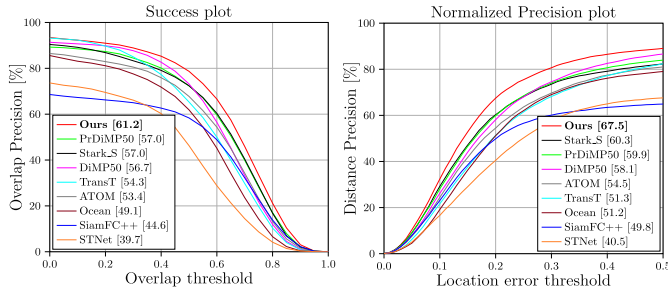

Fig. 9. Success plot and normalized precision plot of similar object sequences on FE240 dataset [18]. The proposed tracker demonstrates superior tracking performance compared to other algorithms in similar object scenarios.
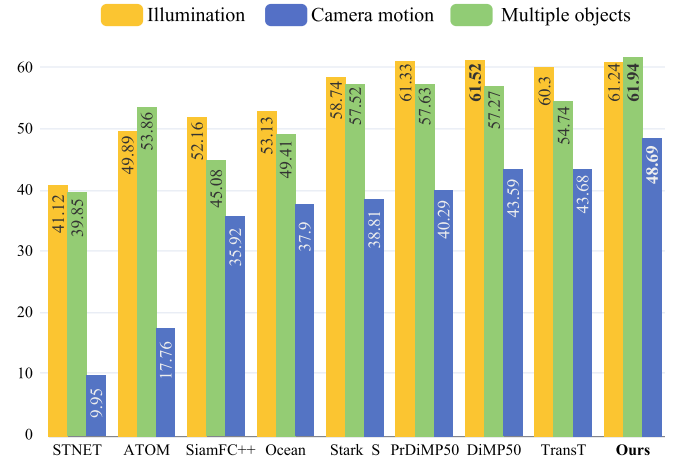

Fig. 10. The AUC metric for camera motion sequences, multiple objects and different illumination sequences on FE240 dataset [18]. Our tracker leads superior performance in these three challenging scenarios. which evidence that the proposed pipeline is suitable and efficient for event-camera based visual object tracking.

TABLE II
LOSS WEIGHT STUDY FOR THE PROPOSED TRACKER

	λ_{iou}	λ_{L_1}	AUC↑	OP_{50} ↑	OP_{75} ↑	Prec.↑	Norm Prec.↑
A.	1	2	55.49	67.75	24.11	87.54	56.16
B.	1	1	55.97	69.92	24.77	87.50	58.06
C.	2	1	56.59	**70.50**	24.10	88.91	**58.57**
Ours	1	5	**56.86**	69.94	**25.03**	**89.19**	58.37

TABLE III
TRACKING SPEED AND ACCURACY COMPARISON FOR THE TOP 4 TRACKERS

Method	TransT [29]	PrDiMP50 [46]	DiMP50 [45]	DANet
AUC(%) ↑	53.46	53.43	54.49	56.86
FPS ↑	50	25	43	**88**

our tracker still outperforms other trackers for most evaluation metrics. In particular, the proposed tracker obtains 40.06%, 55.88%, and 55.47% in terms of AUC, Precision, and Normalized Precision, which is described in Table I. Since STNet [19] needs to split an event frame into 5 parts, we failed to test on the dataset as some raw events are lost. We also report the success plot and normalized precision plot in Figure 7. The visualization of tracking similar objects and fast motion blur scenes can be seen on the second and fourth row of Figure 11, where our tracker can keep tracking the target person precisely and also demonstrate excellent tracking performance in fast motion blur scenes. More visual examples are shown in Figure 6 line 3-4.

c) Comparison on EED dataset: We present the performance comparison of target tracking metrics on the EED dataset in Table I, demonstrating that our proposed DANet achieves favorable performance across multiple metrics. It is noteworthy that the targets in this dataset have smaller sizes and shorter sequence lengths and the sensor used to capture this dataset is older. Furthermore, the data annotation in the

dataset is not as accurate as that in the FE240 dataset [18] and VisEvent dataset [21]. Nevertheless, these experimental results still demonstrate the tracker's robustness on event cameras of different models. More visualizations are illustrated in Figure 5 line 5-6.

d) Loss weights: To find the optimal loss weights, as shown in Table II, we try different loss weight combinations for IOU loss and L1 loss. According to the results, the best weight combination for λ_{IOU} and λ_{L_1} is 1:5, which surpasses the other weight combinations to a great extent.

e) Efficiency: It takes about 7 hours to train DANet and the tracker is able to perform inference at 88 FPS on a single V100 GPU. We report the tracking speed and AUC for the top 4 trackers in Table III. Our tracker surpasses other two trackers in both speed and accuracy, which indicates the efficiency of the proposed method.

C. Ablation Study

We are motivated that event frame can be used to not only distinguish target objects but also help to introduce motion cues to help find moving objects. Thus, our ablation mainly focuses on the following parts: (a) The impact of the
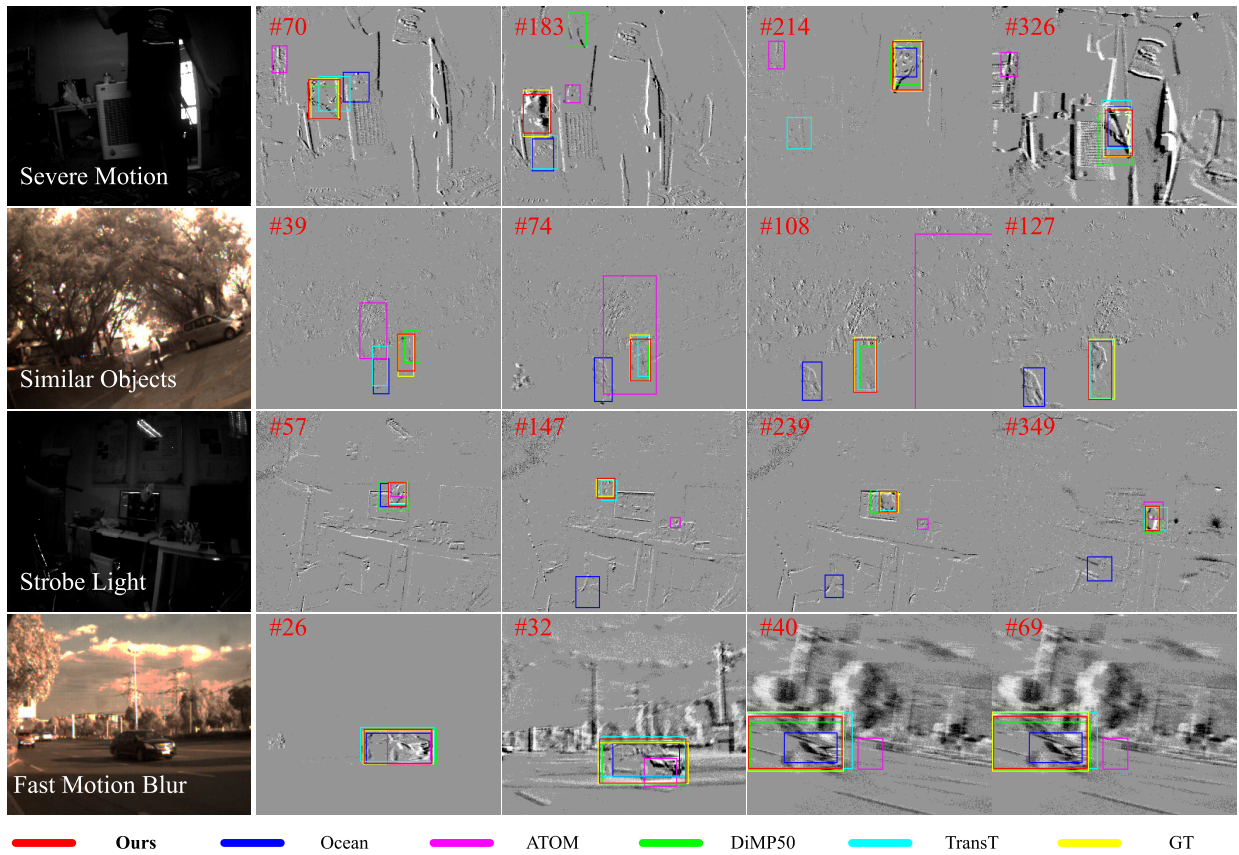
Fig. 11. Four examples to evidence the distractor-aware ability of our proposed tacker against other state-of-the-art trackers on FE240 dataset [18] and VisEvent dataset [21]. The first sequence is a severe camera motion sequence, where background noise can severely distract the moving target. The second sequence is a static camera but multiple person are moving together, which becomes a challenge for movement identification. The images in first column are only used for visualization. We can see that the proposed method can still performs well in camera motion scenario and multiple co-moving object scenario. The third and fourth columns are strobe light and fast motion blur scenes, the proposed tracker still exhibits a considerable advantage in terms of performance in both of these scenarios.

TABLE IV

QUANTITATIVE ABLATION RESULTS THAT DEMONSTRATE THE EFFECTIVENESS OF OUR PROPOSED MODULES ON THE **FE240** DATASET. ✓MEANS THE MODULE IS FULLY USED, "SA" DENOTES THE SELF-ATTENTION IN EACH MODULE. "ECA+CFA" IS THE PROPOSED FUSION BLOCK USED IN [29].TFM(S) DENOTES ADD A SHORTCUT INTO TFM

	TAN	MAN	Integration	AUC↑	OP$_{50}$ ↑	OP$_{75}$ ↑	Prec.↑	Norm Prec.↑	Params.	FLOPs
A.	w/o decoder		N/A	55.59	68.48	25.29	86.52	56.58	4.30M	3.27G
B.		✓	N/A	53.00	64.12	19.75	85.81	52.58	3.71M	2.95G
C.	w/o SA	✓	TFM	55.91	68.84	24.36	87.64	57.29	7.07M	3.61G
D.	✓	w/o SA	TFM	53.40	66.15	22.23	84.39	54.20	7.33M	3.65G
E.	w/o SA	w/o SA	TFM	55.77	69.35	23.54	87.59	57.16	6.93M	3.52G
F.	✓		TFM	54.96	67.23	24.35	86.26	55.15	6.28M	3.30G
G.	w/o decoder	✓	ADD	56.53	**69.96**	25.14	88.58	**58.60**	5.48M	3.70G
H.	N/A	N/A	ECA+CFA [29]	55.93	69.75	24.06	87.50	57.14	5.75M	3.86G
I.	✓	✓	TFM	56.32	69.47	25.26	88.12	57.44	7.46M	3.73G
Ours	✓	✓	TFM(S)	**56.86**	69.94	25.03	**89.19**	58.37	7.46M	3.73G

proposed TAN and MAN; (b) The benefits of self-attention modules in each proposed network; (c) The efficiency of the fusion mechanism; (d) The impact of different feature extractors. (e) The event representation. The following experiments are carried out on the FE240 dataset [18]. All the experiment results in Table IV have shown the superior performance of our proposed tracking pipeline for event camera.

a) Module: As introduced in Section III, the target-aware network aims to match the template frame and the search frame to propose target embeddings, while the motion-aware network aims to extract moving object features. Thus, we evaluate different network structures elaborated below. (*A*) we remove the decoder of TAN, and the enhanced feature map is then sent to the regression head to predict the final bounding box. We notice the performance degradation by 1.27%,

2.67% and 1.79% in terms of AUC, precision, and normalized precision.(*B*) we remove the TAN, so the information from the template frame does not interact with the search frame. We observe a big performance degradation of 3.86%. Note that the search frame keeps updating according to the latest tracking results. As MAN only provides object motion information, which degrades our model to be a detector, and thus leads to a significant performance drop. Nonetheless, the tracking performance still outperforms many trackers in Table I.

b) Importance of self-attention: We also tried to find the importance of the self-attention module in transformer block. The experiments below use both the TAN and MAN, the difference is whether the self-attention is applied in each network. (*C*) we remove the self-attention units in TAN; (*D*) we remove the self-attention units in MAN, and (*E*) we remove the self-attention units in both TAN and MAN. Compared with our final configuration, we can see that the self-attention module helps to improve the tracking performance without introducing too many parameters and computational costs.

c) Fusion mechanism: To validate the efficiency of our fusion mechanism, we measure and compare the performance of different structures. The TFM denotes use TFM but removes the shortcut come from TAN, and TFM(S) denotes use the shortcut. (*F*) we remove the MAN and fuse the encoder output and decoder output of TAN, which leads to poor performance. (*G*) we replace the attention-based fusion by concatenating the outputs of TAN encoder and MAN. We observe that the model performance decreases by 0.33% and 0.61% in terms of AUC and precision. We also evaluate the fusion mechanism proposed in [29], which also uses transformer to interact with features from target frame and search frame. (*I*) we use TFM without the shortcut from TAN, thus the features from the encoder of TAN are wasted. We show the superior performance of the tracker according to the results, which indicates the fusion model contributes to the overall performance.

d) Impact of different feature extractors: We further investigate the impact of different backbones and we show the comparison results in Table V, including ResNet [52] with varied depths, ResNext [63], AlexNet [64], and VGG19 [65]. For ResNet [52], VGG19 [65] and ResNext [63]. We prune the later layers of these networks and applied 1×1 convolution before inputting them into TAN and MAN to accommodate the input size requirements of the subsequent Transformer Encoder structure. According to the results, except for AlexNet [64], the complexity of the feature extractors does not make much difference to the tracker performance but it can significantly influence the model size and the computational cost, so we only use ResNet-18 [52] as our final feature extractor, which has the lowest parameters and computation costs but still archive the best performance in AUC, precision and normalized precision, respectively.

e) Impact of event representation: We carry several experiments to find suitable event representations. On the one hand, For a fixed time interval, we use event image [66] and voxel grid [67] to form the input of the proposed tracker, the results is shown in Table VI, we can see the used event representation performs best across most metrics. On the

TABLE V
COMPARISON FOR THE PROPOSED TRACKER WITH DIFFERENT FEATURE EXTRACTORS

Backbones	AUC↑	Prec. ↑	Norm Prec. ↑	Params.	FLOPS
AlexNet [64]	48.38	76.64	48.91	14.82M	10.49G
VGG19 [65]	56.02	87.38	57.01	25.19M	43.70G
ResNet-34 [52]	56.85	88.42	59.22	8.13M	5.24G
ResNet-50 [52]	55.43	87.41	56.82	18.81M	11.20G
ResNext-50 [63]	55.73	87.20	58.07	18.78M	11.27G
ResNet-18 [52]	**56.86**	**89.19**	**59.37**	**7.46M**	**3.73G**

TABLE VI
EVENT REPRESENTATION STUDY FOR THE PROPOSED TRACKER

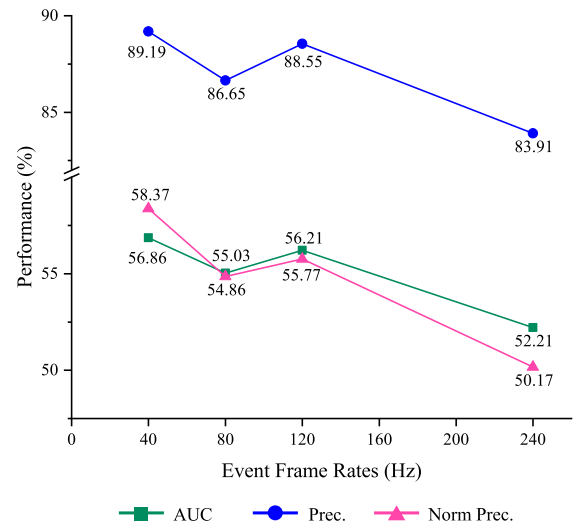| Representation	AUC↑	OP_{50} ↑	OP_{75} ↑	Prec.↑	Norm Prec.↑
Event Frame [66]	54.20	66.74	22.11	86.12	54.80
Voxel Grid [67]	55.94	68.22	**25.72**	87.54	55.70
Ours	**56.86**	**69.94**	25.03	**89.19**	**58.37**

Fig. 12.   Performance of various event frame rates on our tracker.

other hand, since the FE240 dataset provide up to 240Hz annotation, we can use different time windows with the used event representation. The results can be seem in Figure 12, which indicates that 40Hz works best for the proposed tracker. We notice a performance drop with the increments on event frame rate, we believe this is due to the fact that the automatic annotation of the FE240 dataset [18] exhibits lower accuracy as the frame rate increases. Additionally, the event data used as input to the network becomes sparser at higher frame rates, which also affects the performance of the tracker.

D. Feature Visualization

Figure 13 shows an example of feature maps from FE240 dataset [18]. The images from left to right are grayscale image, the cropped and padded event frame, the output feature maps w/o and w/ MAN for the last convolution layer of the center regression branch. We can observe that using the MAN improves object localization.

E. Fuse Event and Frame for Robuster Tracking

Since FE240 dataset provides aligned event stream and grayscale frame, we can jointly utilize the information from
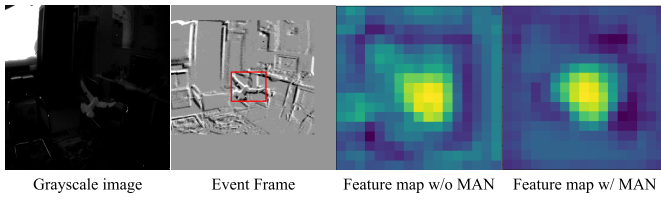
Fig. 13. The feature map output from the last convolution blocks. The crop and padded event search frame and the target are visualized in the second column.

TABLE VII
STUDY FOR FUSING RGB DOMAIN AND EVENT DOMAIN BASED ON THE PROPOSED TRACKER

TAN	MAN	AUC↑	Prec.↑	Norm Prec.↑
Frame	Frame	38.53	64.28	34.44
Event	Event	56.86	89.19	58.37
Frame + Event	Frame + Event	56.24	86.98	57.82
Frame + Event	Event	57.32	**90.74**	57.55
Frame	Frame + Event	**57.87**	89.33	**61.57**

event domain and RGB domain. In this experiment, we use add operator to fuse features and design several fusion strategies based on the proposed pipeline. The fusion operation is performed after the Feature Extract Network. Table VII shows the results for different fusion methods. As the RGB frames suffer from HDR and fast-moving scenes, the performance of a tracker relying exclusively on RGB images is significantly compromised. We further investigate that employing RGB features as inputs for the Target-Aware Network (TAN) and integrating RGB features with event features as inputs for the Motion-Aware Network (MAN) effectively enhances the tracking performance through multimodal fusion. Compared to event only tracker, this fusion approach results in improvements of 1.01%, 0.14%, and 3.20% in AUC, Precision, and Normalized Precision metrics, respectively. We attribute this improvement to the rich target texture information present in RGB images, which significantly improves the target discrimination capability of the TAN network.

F. Discussion

a) Limitation: The performance of the proposed offline tracker primarily relies on the quality of the search event frames. As shown in Figure 14, when there is adequate events of target, the tracker can effectively locate the target. However, in cases where the events of the tracked object are sparse, target discrimination extract from TAN and motion feature extract from MAN are unreliable, leading to tracking failures and potential degradation of subsequent tracking performance. One possible approach is to assign a confidence score to the objects in the search event frames and multiply this score with the regression box. In this way, when the score is high, the tracking result can be given high confidence, while for low scores, the output tracking box will be restricted around the previous location.

b) Future work: The proposed tracking method is designed for event-camera based visual object tracking. However, we do not leverage the asynchronous natural for event camera. Inspired by [68], a promising future research direction is to treat the raw input events as 3D point cloud and use
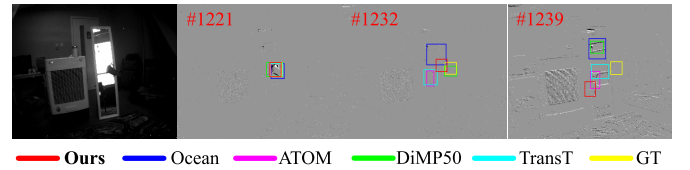

Fig. 14. Illustration of failure cases. Since the object generate very few events in the search frame, there is not enough object features computed in the proposed network, as a result, the tracker can not identify the target in these frames and outputs imprecise bounding boxes.

PointNet++ [69] based method or graph-based method such as [70] to extract spatial-temporal features. As event stream can be treated to spike train, another strategy is to use Spiking Neural Network (SNN) [71] to process event data. Several works have managed to use Conv-SNN to process and propagate the event frame for vision tasks [72], [73], [74], and there is still a promising direction on training the pure Spiking Neural Network for visual tracking with the asynchronous events as input.

V. CONCLUSION

In this paper, we proposed an event-based tracker that is based on a Siamese network structure and introduces a motion-aware network and target-aware network to exploit both motion cues and edge cues from the event camera, and use Target-motion Fusion module to fuse the features for regressing the target coordinates. We used two public event-based tracking dataset to validate the efficiency of our proposed tracker, the extensive validation and ablation experiences showed that our work has a significant performance promotion, and the inference speed is much faster than other online trackers. Our future work will focus on modeling the motion pattern for target movements and making use of the asynchronous feature of event data.

REFERENCES

[1] G. Gallego et al., "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, Jan. 2022.

[2] Y. Deng, H. Chen, H. Chen, and Y. Li, "Learning from images: A distillation learning framework for event cameras," *IEEE Trans. Image Process.*, vol. 30, pp. 4919–4931, 2021.

[3] J. Li, J. Li, L. Zhu, X. Xiang, T. Huang, and Y. Tian, "Asynchronous spatio-temporal memory network for continuous event-based object detection," *IEEE Trans. Image Process.*, vol. 31, pp. 2975–2987, 2022.

[4] N. J. Sanket et al., "EVDodgeNet: Deep dynamic obstacle dodging with event cameras," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 10651–10657.

[5] D. Falanga, K. Kleber, and D. Scaramuzza, "Dynamic obstacle avoidance for quadrotors with event cameras," *Sci. Robot.*, vol. 5, no. 40, Mar. 2020.

[6] C. Scheerlinck, H. Rebecq, D. Gehrig, N. Barnes, R. E. Mahony, and D. Scaramuzza, "Fast image reconstruction with an event camera," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 156–163.

[7] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "High speed and high dynamic range video with an event camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 1964–1980, Jun. 2021.

[8] L. Pan, C. Scheerlinck, X. Yu, R. Hartley, M. Liu, and Y. Dai, "Bringing a blurry frame alive at high frame-rate with an event camera," in *Proc. CVPR*, Jun. 2019, pp. 6813–6822.

[9] P. R. G. Cadena, Y. Qian, C. Wang, and M. Yang, "Spade-E2Vid: Spatially-adaptive denormalization for event-based video reconstruction," *IEEE Trans. Image Process.*, vol. 30, pp. 2488–2500, 2021.

[10] H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3D reconstruction and 6-DoF tracking with an event camera," in *Proc. ECCV*, 2016, 349–364.

[11] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM," *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 142–149, Feb. 2017.

[12] J. Jiao, H. Huang, L. Li, Z. He, Y. Zhu, and M. Liu, "Comparing representations in tracking for event camera-based SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 1369–1376.

[13] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 994–1001, Apr. 2018.

[14] Y. Zhou, G. Gallego, and S. Shen, "Event-based stereo visual odometry," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1433–1450, Oct. 2021.

[15] F. Barranco, C. Fermuller, and E. Ros, "Real-time clustering and multi-target tracking using event-based sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 5764–5769.

[16] A. Glover and C. Bartolozzi, "Robust visual tracking with a freely-moving event camera," in *Proc. IROS*, Sep. 2017, 3769–3776.

[17] H. Chen, D. Suter, Q. Wu, and H. Wang, "End-to-end learning of object motion estimation from retinal events for event-based object tracking," in *Proc. AAAI*, 2020, pp. 10534–10541.

[18] J. Zhang, X. Yang, Y. Fu, X. Wei, B. Yin, and B. Dong, "Object tracking by jointly exploiting frame and event domain," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13023–13032.

[19] J. Zhang et al., "Spiking transformers for event-based single object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8791–8800.

[20] Y. Zheng, Z. Yu, S. Wang, and T. Huang, "Spike-based motion estimation for object tracking through bio-inspired unsupervised learning," *IEEE Trans. Image Process.*, vol. 32, pp. 335–349, 2023.

[21] X. Wang et al., "VisEvent: Reliable object tracking via collaboration of frame and event flows," 2021, *arXiv:2108.05015*.

[22] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. ECCV*, 2018, pp. 294–310.

[23] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, "Learning texture transformer network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5790–5799.

[24] H. M. Kasem, K.-W. Hung, and J. Jiang, "Spatial transformer generative adversarial network for robust image super-resolution," *IEEE Access*, vol. 7, pp. 182993–183009, 2019.

[25] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. ECCV*, 2020, pp. 213–229.

[26] Z. Sun, S. Cao, Y. Yang, and K. Kitani, "Rethinking transformer-based set prediction for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3591–3600.

[27] Y. Fang et al., "You only look at one sequence: Rethinking transformer in vision through object detection," in *Proc. NeurIPS*, 2021, pp. 1–15.

[28] N. Zhang, J. Han, and N. Liu, "Learning implicit class knowledge for RGB-D co-salient object detection with transformers," *IEEE Trans. Image Process.*, vol. 31, 4556–4570, 2022.

[29] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking is the process of continuously monitoring the key performance indicators (KPIs) of transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8122–8131.

[30] N. Wang, W. Zhou, J. Wang, and H. Li, "Transformer meets tracker: Exploiting temporal context for robust visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1571–1580.

[31] Z. Cao, C. Fu, J. Ye, B. Li, and Y. Li, "HiFT: Hierarchical feature transformer for aerial tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15437–15446.

[32] A. Vaswani et al., "Attention is all you need," in *Proc. NIPS*, 2017, pp. 1–11.

[33] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1328–1338.

[34] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, Jul. 2014.

[35] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, pp. 1–45, 2006.

[36] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel, "A survey of appearance models in visual object tracking," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 4, pp. 1–48, Sep. 2013.

[37] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional Siamese networks for object tracking," in *Proc. ECCV*, 2016, pp. 850–865.

[38] K. Dai, D. Wang, H. Lu, C. Sun, and J. Li, "Visual tracking via adaptive spatially-regularized correlation filters," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4665–4674.

[39] P. Li, B. Chen, W. Ouyang, D. Wang, X. Yang, and H. Lu, "GradNet: Gradient-guided network for visual object tracking," in *Proc. CVPR*, 2019, pp. 6161–6170.

[40] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4293–4302.

[41] Z. Zhang and H. Peng, "Deeper and wider Siamese networks for real-time visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4586–4595.

[42] F. Liu, C. Gong, X. Huang, T. Zhou, J. Yang, and D. Tao, "Robust visual tracking revisited: From correlation filter to template matching," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2777–2790, Jun. 2018.

[43] W. Zhou, L. Wen, L. Zhang, D. Du, T. Luo, and Y. Wu, "SiamCAN: Real-time visual tracking based on Siamese center-aware network," *IEEE Trans. Image Process.*, vol. 30, pp. 3597–3609, 2021.

[44] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Atom: Accurate tracking by overlap maximization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4655–4664.

[45] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6181–6190.

[46] M. Danelljan, L. Van Gool, and R. Timofte, "Probabilistic regression for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7181–7190.

[47] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning spatio-temporal transformer for visual tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10428–10437.

[48] A. Mondal, R. Shashant, J. H. Giraldo, T. Bouwmans, and A. S. Chowdhury, "Moving object detection for event-based vision using graph spectral clustering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 876–884.

[49] J. Zhang et al., "Frame-event alignment and fusion network for high frame rate tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023.

[50] Z. Zhu, J. Hou, and D. O. Wu, "Cross-modal orthogonal high-rank augmentation for RGB-event transformer-trackers," 2023, *arXiv:2307.04129*.

[51] C. Tang et al., "Revisiting color-event based tracking: A unified network, dataset, and metric," 2022, *arXiv:2211.11010*.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[53] A. Nibali, Z. He, S. Morgan, and L. Prendergast, "Numerical coordinate regression with convolutional neural networks," 2018, *arXiv:1801.07372*.

[54] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 658–666.

[55] A. Mitrokhin, C. Fermüller, C. Parameshwara, and Y. Aloimonos, "Event-based moving object detection and tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1–9.

[56] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.

[57] L. Huang, X. Zhao, and K. Huang, "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1562–1577, May 2021.

[58] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[59] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. NIPS*, 2019, pp. 1–12.

[60] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, 2018, pp. 1–16.

[61] Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu, "SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines," in *Proc. AAAI*, 2020, pp. 12549–21556.

[62] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu, "Ocean: Object-aware anchor-free tracking," in *Proc. ECCV*, 2020, pp. 771–787.

[63] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.

[64] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NeurIPS*, 2012, pp. 1–9.

[65] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[66] H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization," in *Proc. Procedings Brit. Mach. Vis. Conf.*, 2017, pp. 1–12.

[67] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 989–997.

[68] Y. Sekikawa, K. Hara, and H. Saito, "EventNet: Asynchronous recursive event processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3882–3891.

[69] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. NeurIPS*, 2017, pp. 1–10.

[70] Z. Zhu, J. Hou, and X. Lyu, "Learning graph-embedded key-event back-tracing for object tracking in event clouds," in *Proc. NIPS*, 2022, pp. 1–15.

[71] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *Frontiers Neurosci.*, vol. 12, p. 331, May 2018.

[72] C. Lee, A. K. Kosta, A. Z. Zhu, K. Chaney, K. Daniilidis, and K. Roy, "Spike-FlowNet: Event-based optical flow estimation with energy-efficient hybrid neural networks," in *Proc. ECCV*, 2020, pp. 366–382.

[73] M. Gehrig, S. B. Shrestha, D. Mouritzen, and D. Scaramuzza, "Event-based angular velocity regression with spiking networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 4195–4202.

[74] H. C. Duwek, A. Shalumov, and E. E. Tsur, "Image reconstruction from neuromorphic event cameras using Laplacian-prediction and Poisson integration with spiking and artificial neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 1333–1341.

Wenxi Liu is currently a Professor with the College of Computer and Data Science, Fuzhou University, Fuzhou, China. His research interests include crowd analysis, segmentation, and visual tracking.

Yuanchen Wang received the bachelor's degree in computer science and technology from Yunnan University in 2020. He is currently pursuing the master's degree with the Department of Computer Science, Dalian University of Technology, China. His research interests include spiking neural networks and memory neural systems.
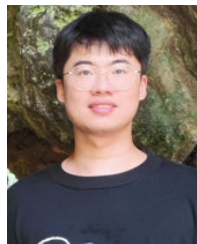
Jiqing Zhang received the B.Eng. degree in computer science and technology from Dalian Maritime University, Dalian, China, in 2017. He is currently pursuing the Ph.D. degree in computer application technology with the Dalian University of Technology, Dalian. His research interests include computer vision, machine learning, and image processing.

Baocai Yin (Member, IEEE) received the B.S. and Ph.D. degrees in computer science from the Dalian University of Technology. He is a Professor of computer science with the Beijing University of Technology, Beijing, China. His research concentrates on digital multimedia and computer vision.

Yingkai Fu received the master's degree in computer science from the Dalian University of Technology, Dalian, China, in 2023. His research interests include event-based vision and robotic vision.

Xiaopeng Wei received the Ph.D. degree from the Dalian University of Technology in 1993. He is a Professor with the Dalian University of Technology. His research areas include computer animation, computer vision, robots, and intelligent CAD.

Meng Li is a Senior Engineer with HiSilicon (Shanghai) Technologies Company Ltd. He has employed for ten years and has dedicated to research in computational vision, image signal processing, and multimodal vision.

Xin Yang received the B.S. degree in computer science from Jilin University in 2007 and the Ph.D. degree in July 2012. He is a Professor with the Department of Computer Science, Dalian University of Technology, China. From 2007 to June 2012, he was a joint Ph.D. Student in graphics with Zhejiang University and UC Davis. His research interests include computer graphics and robotic vision.