# Motion Robust High-Speed Light-Weighted Object Detection with Event Camera

Bingde Liu, Chang Xu, Wen Yang, Huai Yu, Lei Yu



***Abstract*—The event camera asynchronously produces the event stream with a high temporal resolution, discarding redundant visual information and bringing new possibilities for moving object detection. Nevertheless, the existing object detectors cannot make the most of the spatial-temporal asynchronous nature and high temporal resolution of the event stream. For one thing, existing methods fail to consider objects with different velocities relative to the event camera's motion, resulting from the global synchronized time window with the whole spatial slice. For another, most of the existing methods rely on heavy models and boost the detection performance with low frame rates, failing to utilize the high temporal resolution characteristic of the event stream. In this work, we propose a motion robust and high-speed detection pipeline which better leverages the event data. First, we design an event stream representation called Temporal Active Focus (TAF), which efficiently utilizes the spatial-temporal asynchronous event stream, constructing event tensors robust to object motions. Then, we propose a module called the Bifurcated Folding Module (BFM), which encodes the rich temporal information in the TAF tensor at the input layer of the detector. Following this, we design a high-speed lightweight detector called Agile Event Detector (AED) plus a simple but effective data augmentation method, to enhance the detection accuracy and reduce the model's parameter. Experiments on two typical real-scene event camera object detection datasets show that our method is competitive in terms of accuracy, efficiency, and the number of parameters. By classifying objects into multiple motion levels based on the optical flow density metric, we further illustrated the robustness of our method for objects with different velocities relative to the camera. The codes and trained models are available at https://github.com/HarmoniaLeo/FRLW-EvD.**

***Index Terms*—Event Camera, Object Detection, Event Representation, Fast-moving Objects, Light Weight Detector**

## I. Introduction

Real-world object detection tasks have extremely high requirements on the detection speed and the robustness of detectors for bad weather, extreme lighting conditions, and fast-moving objects. Autonomous driving is one of its typical application scenarios which is demanding for safety and robustness. Therefore, diverse sensors are leveraged to obtain robust detection under various conditions. Among them, traditional frame-based RGB cameras [1], [2] are used to provide rich semantic and texture information about the surroundings. The LIght Detection And Ranging (LIDAR) sensor-based object detectors [3]–[6] are employed to compensate for RGB cameras' failure under extreme lighting conditions. In addition, the millimeter-wave radar is utilized to enhance the detection performance under adverse weather conditions [7]. Despite these progresses, none of these sensors are particularly good at seeing fast-moving objects [2], [8] (*e.g.* the sudden appearing pedestrian in front of a car), thus degrading the detection accuracy, which leads to a safety hazard.

As a new type of vision-based sensor [9], the event camera has many advantages, including high temporal resolution, large dynamic range, and reduction of redundant information [10]–[13]. Benefiting from these properties, event cameras have great potential for applications in scenarios where most sensors (*e.g.* RGB camera, LIDAR, and millimeter-wave radar) are subject to motion blur and fast-moving objects, with very low energy consumption and small data storage costs [14], [15].

So far, there have been some works that apply event cameras to object detection tasks, mainly in the field of autonomous driving [16]–[18]. The contributions of the event cameras to the field are their ability to supplement other sensors, provide concurrent streams of temporal contrast events, and offer several advantages over traditional cameras.

Among the works, the methods with leading accuracy levels (*e.g.,* YOLE [19], RED [20], and ASTMNet [21]) follow the paradigm of a global time window (cuboid-shaped shadow region in Fig. 1) for event representation and a deep neural network-based object detector with large amounts of parameters to boost detection accuracy, their general structure is shown in Fig. 1. For example, the state-of-the-art ASTMNet [21] deeply explores this paradigm by adaptively adjusting the duration of the global time window and leveraging the recurrent-convolutional architecture.

Despite the exploration of adaptive and asynchronous temporal sampling, there still exist two main drawbacks in existing methods, namely the global time window and the unsatisfactory running speed [19]–[24]. Firstly, the asynchronous exploration in existing methods is limited by the global time window, which maintains the whole spatial slice when adjusting the temporal duration. Hence, existing methods cannot take into account objects with different velocities relative to the event camera. When there are multiple objects in the field of view at the same time, a long time window may lead to the occurrence of motion blur for fast-moving objects, while a short time window may be unable to gather enough information for slow-moving objects. Secondly, current object

Manuscript received 8 January 2022; revised 5 March 2023; accepted 5 April 2023. Date of publication XXX 2023; date of current version XXX 2023. This work was supported in part by the National Natural Science Foundation of China under Grants 62271354; in part by the Natural Science Foundation of Hubei Province, China under Grants 2022CFB600 and 2021CFB467. The Associate Editor coordinating the review process was Dr. Emanuele Zappa. (Corresponding author: Wen Yang.)

The authors are with the School of Electronic Information, Wuhan University, Wuhan 430072, China. (e-mail: {liubingde, xuchangeis, yangwen, yuhuai, ly.wd}@whu.edu.cn)

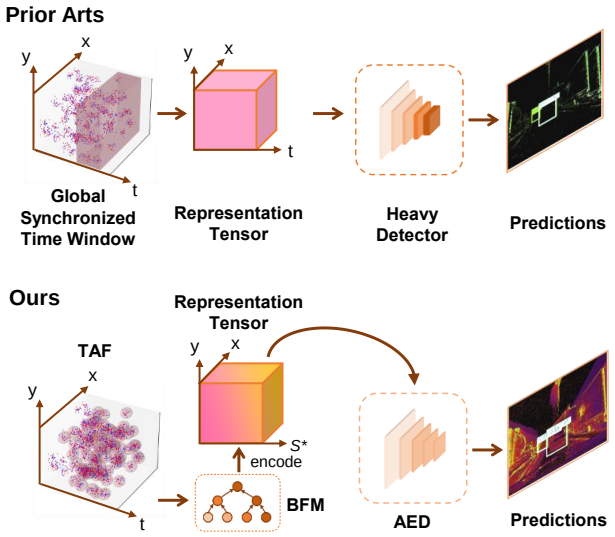



Fig. 1: Structure of our approach and comparison with previous approaches [20]–[22], [24]. The TAF, BFM, and AED stand for the temporal active focus representation, bifurcated folding module, and agile event detector, respectively. $S^*$ denotes the semantic dimension encoded via the BFM from the temporal dimension. Unlike previous methods, our TAF approach uses an asynchronous approach to sample the event stream and uses BFM to extract semantics from the temporal dimension when constructing the representation tensor. Our AED detector is also more lightweight.

detectors require a large number of model parameters to achieve high accuracy, which implies a large computational burden and low inference speed. Applying those algorithms to event data does not match its high temporal resolution nature.

To solve the above problems, we propose a motion-robust and high-speed object detection pipeline for event data. First of all, we design a new event stream representation called Temporal Active Focus (TAF), to conquer the drawbacks of the global time window. It can better take advantage of the asynchronous nature of event data, spatially and temporally. By asynchronously sampling the Event Measurement Field [25] in a low computationally demanding manner, it constructs tensors containing different time range information on different spatial and polar positions. Then, we design a module called Bifurcated Folding Module (BFM), cooperating with the TAF. The BFM encodes the rich temporal information in TAF tensors before feeding the tensors into the detector, improving the detection accuracy. Moreover, we propose a lightweight detector called Agile Event Detector (AED), which has a high inference speed that better matches the event data's high temporal resolution. In addition, to improve the generalization capability of the detector, we propose to use a simple but effective data augmentation strategy including random flipping and cropping. The overall architecture of our approach and comparison with the previous approach is shown in Fig. 1.

We choose two typical real-scene event camera object detection datasets for experiments: the complete Prophesee GEN1 Automotive Detection Dataset (GEN1 Dataset) [26] and the Prophesee 1 MEGAPIXEL Automotive Detection Dataset [20] with partial annotation (1 MEGAPIXEL Dataset (Subset)). We measure the motion speed of objects in the datasets by computing optical flow, and then we classify them into 5 different motion levels. Experiments show that compared with the current state-of-the-art methods, our method has a far lower number of parameters and much higher running speed while retaining competitive accuracy. The experiments also show that our method has high detection accuracy under all 5 motion levels, which demonstrates motion robustness.

In summary, our contributions are as follows:

1) We propose an event stream representation method called Temporal Active Focus (TAF), which better leverages the asynchronous nature of event stream data, spatially and temporally.
2) We design a module called Bifurcated Folding Module (BFM) to encode the rich temporal information in the TAF tensor.
3) We introduce a high-speed lightweight detector called Agile Event Detector (AED) along with a simple but effective data augmentation method.
4) We conduct experiments on two typical real-scene event camera object detection datasets. The experimental results show that compared with the state-of-the-art methods, our method has a far lower number of parameters and a much higher running speed. Moreover, our method retains competitive accuracy and superior motion robustness.

## II. Related Works

In this section, we present related works, including event representations and event-based object detection methods.

### A. Event representations

Currently, representative approaches mainly process events in batches into tensor-like representations as the input of the object detection algorithms. Much research has demonstrated that the statistics of the event representation tensors overlap with those of natural images [17], [25], [27]. The event representation methods are divided into two main categories: the Event Spike Tensor representation [17], [27]–[29] and the representation of emulating spike neural networks [19], [27], [30]. Event Spike Tensor is a data representation method based on discretizing the Event Measurement Field [25]. The representation of emulating spike neural networks is to let the spatial locations of newly occurring events always have larger values in a tensor.

There are hyper-parameters to be specified in all the existing event representation methods. The representative Event Spike Tensor requires hyper-parameters including the measurement function, the convolution kernel function, and timestamps for executing convolutions. There are also some works [21], [25] that explore the possibility of learning the measurement function or the convolution kernel function directly from

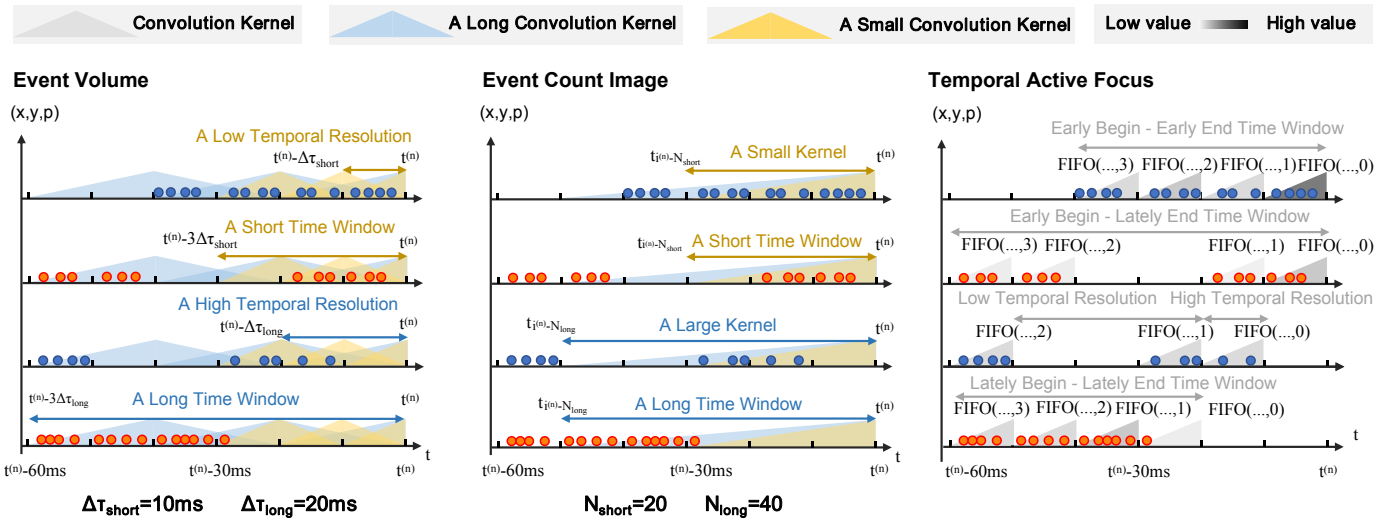



Fig. 2: Comparisons between the process of constructing Event Spike Tensors and the process of TAF at the $n^{th}$ detection. $t^{(n)}$ is the timestamp of the $n^{th}$ detection. $\Delta\tau$ is the sampling period. $B$ is the temporal dimension length. $N$ is the number of events to sample.

the raw event stream, but the timestamps for performing kernel convolutions are still hyper-parameters that need to be specified. The representation of emulating spike neural networks requires a global numerical transformation function as the hyper-parameter to keep the values in the tensor in a value range. The hyper-parameters determine the trade-off between the globally fixed time window length and temporal resolution. Therefore, the existing event representations cannot take into account objects with different velocities relative to the event camera.

Inspired by a queue-based event representation method [31], we propose to explore its possibility in object detection tasks. We believe that its ability to asynchronously sample the event stream data at each spatial and polar position is significant to the object detection tasks. In our work, we propose a new kind of event stream representation method, solving the problem via asynchronously sampling the Event Measurement Field [25] to adjust the time window length and temporal resolution on each spatial and polar position flexibly.

### *B. Event stream object detectors*

Mainstream approaches use Convolutional Neural Networks (CNN) with a large number of parameters to build object detectors [1], [19], [22], [23], [32]. The main problems with these methods are that they are not accurate, light, and fast enough for event stream data object detection tasks. Different from these methods, some approaches use recurrent convolutional neural network object detectors [20], [21] to improve the accuracy. However, the memory mechanisms used in the recurrent convolutional neural networks are not designed according to the characteristics of the event stream data. Therefore, those detectors are expensive to train and slow to run. In our work, we design a high-speed lightweight detector with competitive accuracy. Our detector is more suitable for event stream data object detection tasks.

## III. Problem Statements

A paradigm for the event-based object detection task was defined by Perot et al. [20]. It is briefly modified here to make it more precise and to facilitate the discussion later on:

- Event stream data: for a camera with the picture size height of $H$ and width of $W$, its event stream data is defined as $E := \{e_i = (x_i, y_i, p_i, t_i)\}_{i\in\mathbb{N}}$, where $x_i \in \{0, 1, ..., W-1\}$ and $y_i \in \{0, 1, ..., H-1\}$ are the coordinates, $p_i \in \{0, 1\}$ is the polarity, $t_i \in [0, T_{max})$ is the timestamp, $T_{max}$ is the maximum duration of the event stream record.
- Annotation: $B^* := \{b_j^* = (x_j, y_j, w_j, h_j, l_j, t_j)\}_{j\in\mathbb{N}}$, where $x_j \in \mathbb{R}$ and $y_i \in \mathbb{R}$ are the coordinates of the upper left corner of a bounding box, $w_j \in \mathbb{R}^+$ is the width of the bounding box, $h_j \in \mathbb{R}^+$ is the height of the bounding box, $l_j \in \{0, ..., L\}$ is the category, $t_j$ is the timestamp.
- Detection: $D(S(E^{(n)}, t^{(n)}))$, where $D(.)$ is the detector, $S(.)$ is the event representation method, $n \in \{1, 2, ...\}$ means it is the $n^{th}$ detection on the event stream, $E^{(n)} := \{e_i\}_{t_i\in[0,t^{(n)})}$ is the available event stream for the $n^{th}$ detection, $t^{(n)} \in [0, T_{max})$ denotes the timestamp of the $n^{th}$ detection.

The event representation methods are divided into two main categories: the Event Spike Tensor representation [17], [27]–[29] and the representation of emulating spike neural networks [19], [27], [30].

The Event Spike Tensor methods can sample the Event Measurement Field [25] with convolution kernel to represent rich information of the event stream. Among them, the Event Volume [29] is a typical Event Spike Tensor. It

uses convolution kernel to sample events in the time window $t_i \in [t^{(n)} - B\Delta\tau, t^{(n)})$ at the $n^{th}$ detection, with a sampling resolution inversely proportional to the sampling period $\Delta\tau \in \mathbb{R}^+$ and a fixed temporal dimension length $B \in \mathbb{Z}^+$. The Event Count Image [17], [27], on the other hand, uses a deformable convolution kernel to sample the recent $N \in \mathbb{Z}^+$ events. The kernel size and the time window covered can therefore be dynamically adjusted according to the frequency of events triggered within the recent event stream. However, the globally synchronized characteristic in these above representations has limitations. As illustrated in Fig. 2, for spatial and polar locations that recently trigger events less frequently, a short global synchronized time window cannot aggregate enough information. On the contrary, for spatial and polar locations that recently trigger events more frequently, a long global synchronized time window leads to a decrease in the temporal resolution, making the convolution kernel cover excessive events, resulting in a loss of information.

The methods of emulating spike neural networks do not apply a linear transformation to the Event Measurement Field, but the value of each spatial location is adjusted by discrete decisions during the event generation process. The generated representation tensors always satisfy that spatial locations of the recently generated events maintain larger values. Therefore, this kind of method supports flexible nonlinear temporal resolution adjustment. The surface of Active Events [27], [30] is a typical example of the representation method of emulating spike neural networks. It gives a 2D snapshot of the latest timestamp of the events in the field of view. The elapse from the timestamp of the $n^{th}$ detection $\Delta t \in \mathbb{R}^-$ is then mapped to the value range $(0, 1)$ with a numerical transformation $e^{\lambda\Delta t}$, where $\lambda \in \mathbb{R}^+$ is a hyper-parameter. Fig. 3 shows the mapping when $\lambda$ takes different values. It can be seen that under large $\lambda$, the mapping gives the newly triggered events a larger temporal resolution, while the information of events triggered further than a threshold from now will hardly be retained. Therefore, still, the Surface of Active Events can only consider events in a certain time window. The length of the time window can be increased by decreasing $\lambda$. However, on the other hand, the temporal resolution will be decreased. Therefore, since the hyper-parameter $\lambda$ is globally applied, the Surface of Active Events also faces the problem of globally synchronized characteristics like the Event Spike Tensor methods.

Fig. 4 shows the visualization of different event representation tensors with different hyper-parameters. In summary, it is hard to find a hyper-parameter that takes into account all objects with different motion speeds relative to the camera in the field of view at the same time.

## IV. Method

This section introduces our core methods, including the event representation method Temporal Active Focus (TAF), the Bifurcated Folding Module (BFM) for fully extracting features from the TAF representation tensor, and the high-speed lightweight detector Agile Event Detector (AED).

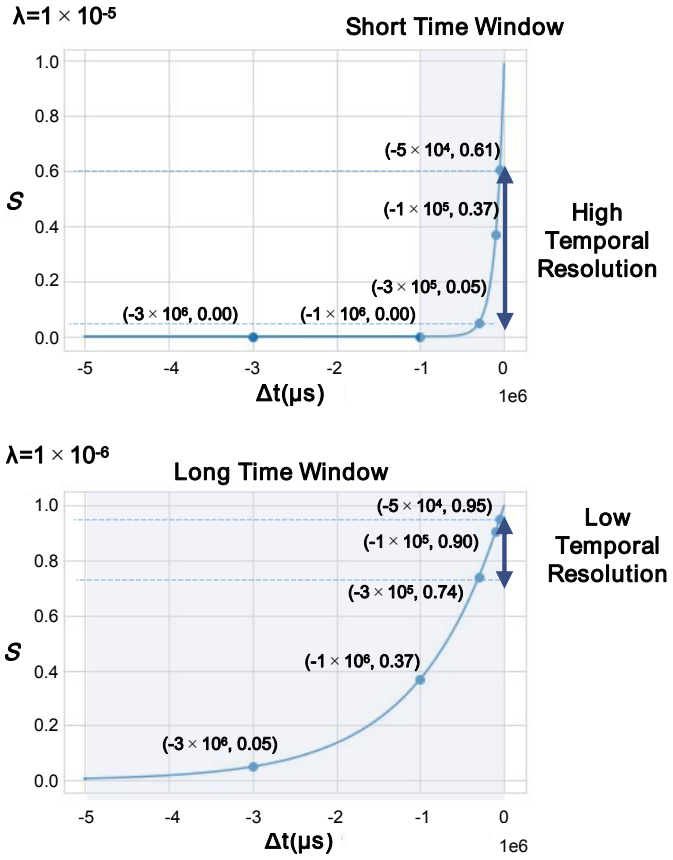



Fig. 3: The mapping between $\Delta t$ and the value in Surface of Active Events tensor under different $\lambda$. $\Delta t$ is the elapse from the timestamp of the $n^{th}$ detection $t^{(n)}$. $\lambda$ is a hyper-parameter.

### A. Temporal Active Focus

Tensors generated by the Temporal Active Focus method can be viewed as a dense version of the Event Spike Tensor. Considering a traditional Event Spike Tensor with variable length in temporal dimension to cover all event stream information by the time for detection, *i.e.*, $B = \lceil \frac{t^{(n)}}{\Delta\tau} \rceil$. Due to the spatial and temporal sparsity of the event stream data [23], such a tensor is also a sparse tensor. Temporal Active Focus (TAF), by contrast, samples the first $K$ latest positions with a non-zero value in the temporal dimension at each spatial and polar position to form a dense tensor.

The infinite-length sparse Event Spike Tensor is impractical to build in terms of time and storage cost. However, since the object detection on the event stream is performed at a certain frequency if the detection period is set to the sampling period $\Delta\tau$, the computational burden can be greatly reduced by using the FIFO queue to incrementally update the TAF tensor. The necessary condition is that the convolution kernel $k(\cdot)$ used for sampling the Event Measurement Field satisfies:

$$\begin{cases} k(x, y, p, t) > 0, & t \in [0, k_{upper}] \\ k(x, y, p, t) = 0, & Others \end{cases} \\ s.t. \quad k_{upper} \in [0, \infty) \tag{1}$$

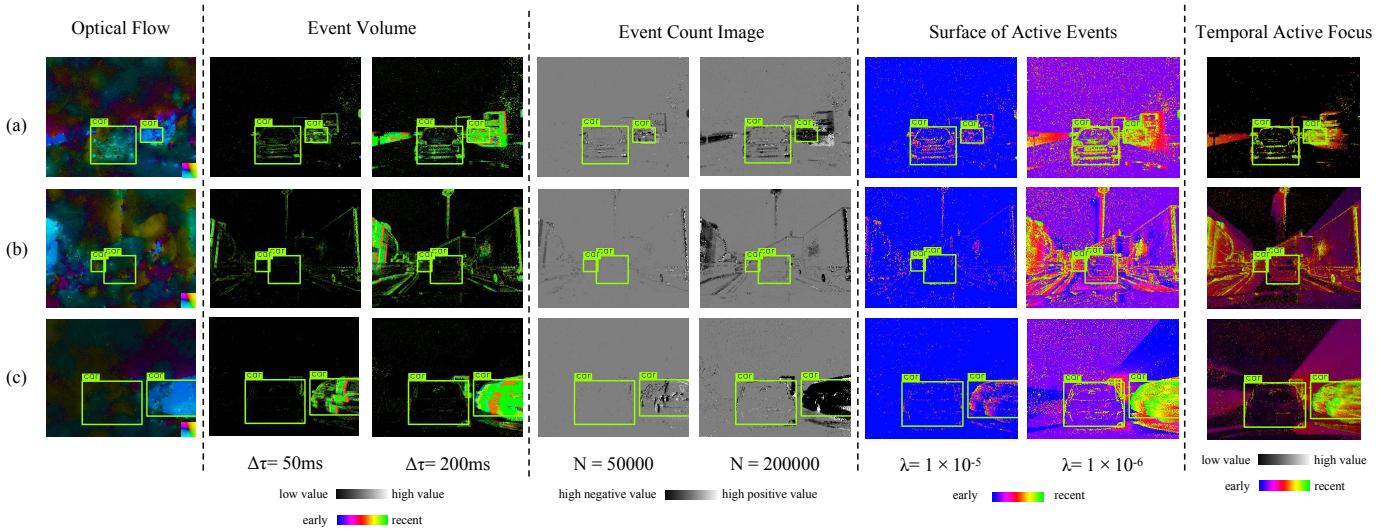



Fig. 4: Visualization of different event representation tensors with different hyper-parameters. Column 1 shows the TV-L1 optical flow plots of the scenes, which are computed by the method proposed by Nagata et al. [33]. In the field of view, (a) is the case where only objects with fast motion relative to the event camera are present, (b) is the case where only objects with slow motion relative to the event camera are present, and (c) is the case where objects with different motion relative to event camera are present at the same time.

TAF maintains a per spatial and polar position specific FIFO queue $FIFO(x, y, p, k)$ with depth $K \in \mathbb{Z}^+$. The queues continuously receive non-zero samples from the event stream, thus generating dense tensor representations. The process of building the TAF tensor $\mathrm{S} \in \mathbb{R}^{2K\times H\times W}$ is shown in Algorithm 1. The convolution kernel $k(\cdot)$ and the measurement $f(\cdot)$ are essential components of Event Spike Tensor as introduced in [25]. For the convolution kernel, we simply adopt the rectangular window function:

$$k(x, y, p, t) = \begin{cases} 1, & t \in [0, \Delta\tau] \\ 0, & Others \end{cases} \tag{2}$$

In order not to lose the absolute position information on the temporal dimension, we use a measurement to calculate the average elapse from the events covered by the convolution kernel to $t^{(n)}$:

$$f^{(n)}(x, y, p, t) = \frac{t^{(n)} - t}{\#\{E^{(n)}_{x,y,p}\}} \tag{3}$$

where $E^{(n)}_{x,y,p} = \{e_i\}_{t_i\in[t^{(n)}-k_{upper},t^{(n)}),x_i=x,y_i=y,p_i=p}$ and $\#(.)$ is the counting symbol.

At the $n$th detection, we have the average time elapses calculated as:

$$\Delta t^{(n)}(E, t, x, y, p) := \sum_{e_i\in E} f(x_i, y_i, p_i, t_i, t^{(n)})k(x - x_i, y - y_i, p - p_i, t - t_i) \tag{4}$$

The non-zero values of the average time elapses will be pushed into the FIFO queues. Then at the $n + 1$th detection, we have new values calculated and pushed into the queues, while old values are updated by: $\Delta t^{(n+1)} \longleftarrow \Delta t^{(n)} + \Delta\tau$.

To limit the range of $\Delta t$ values (in milliseconds), we apply another measurement function based on the logarithmic transformation:

$$F(\Delta t) = 1 - \frac{\ln(1 + \Delta t \times 10^{-4})}{\ln(1 + T_{max} \times 10^{-4})} \tag{5}$$

where $T_{max}$ is the maximum duration of the event stream (in milliseconds). Fig. 5 shows that the values of $F(\cdot)$ on the whole$\Delta t \in [-T_{max}, 0)$ interval all reflect high resolutions.

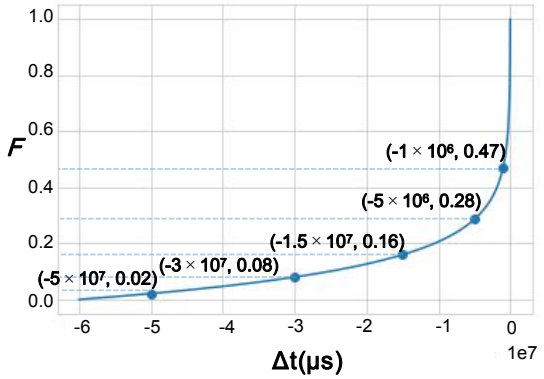



Fig. 5: The mapping between the elapse $\Delta t$ and the value of the measurement function $F(\cdot)$ when the maximum duration of the event stream record $T_{max} = 6 \times 10^7 \mu s$.

Fig. 2 visualizes the asynchronous characteristic of TAF, which shows that TAF can adapt the interval of applying convolutions asynchronously according to the event triggering frequency. Therefore, a larger temporal resolution is applied for periods with a higher event triggering frequency, while a

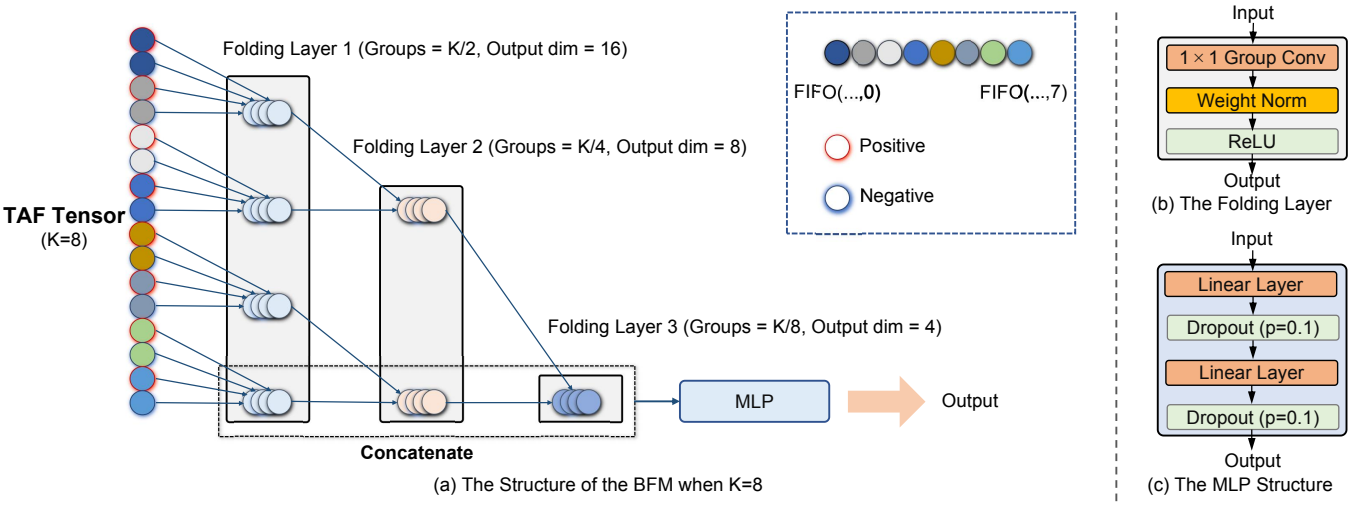



Fig. 6: (a) is the structure of the BFM module. (b) and (c) are the structure of its components. $K$ is the length of the TAF queue.

**Algorithm 1:** The process of building the TAF tensor when continuously performing object detection on the event data stream.

**Input:** $\mathcal{X} = \{0, 1, ..., W-1\}$, $\mathcal{Y} = \{0, 1, ..., H-1\}$, $\mathcal{P} = \{0, 1\}$, $\mathcal{N} = \{1, 2, ..., \lceil \frac{T_{max}}{\Delta\tau} \rceil\}$, $\mathcal{C} = \{0, 1, ..., 2K-1\}$

#Initialize FIFO queues
**for** each $x \in \mathcal{X}$, $y \in \mathcal{Y}$, $p \in \mathcal{P}$ **do**
  Initialize all values in $FIFO(x, y, p)$ with 0;
#Detect on the event flow
**for** each $n \in \mathcal{N}$ **do**
  #Perform convolution
  $E_{sub}^{(n)} := \{e_i\}_{t_i \in [n\Delta\tau - k_{upper}, n\Delta\tau)}$
  #Update FIFO queues
  **for** each $x \in \mathcal{X}$, $y \in \mathcal{Y}$, $p \in \mathcal{P}$ **do**
    **if** $\Delta t^{(n)}(E_{sub}^{(n)}, n\Delta\tau, x, y, p) > 0$ **then**
      Add $\Delta t^{(n)}(E_{sub}^{(n)}, n\Delta\tau, x, y, p)$ to $FIFO(x, y, p)$.
    **for** each $c \in \mathcal{C}$ **do**
      Update $FIFO(x, y, p)[c]$ with new value
  #Construct tensor
  **for** each $x \in \mathcal{X}$, $y \in \mathcal{Y}$, $c \in \mathcal{C}$ **do**
    $S_{c,y,x} := F\{FIFO(x, y, c - 2\lfloor \frac{c}{2} \rfloor)[\lfloor \frac{c}{2} \rfloor]\}$
  **Output:** S $:= \{S_{c,y,x}\}_{c \in \mathcal{C}, y \in \mathcal{Y}, x \in \mathcal{X}}$

smaller temporal resolution is applied otherwise. In addition, the time interval of event streams considered at each spatial and polar position starts and ends flexibly, which means that the time window can also be adjusted asynchronously. The positions and values in the temporal dimension will jointly encode information such as the elapse of the most recently triggered event at each position, the number of short-time triggered events, and the frequency of triggered events in a certain period, thus containing more information. Column 8 in Fig. 4 shows the visualization of the information considered by Temporal Active Focus. It can be seen that the TAF tensor can consider equal high-resolution information for objects with different motion speeds.

### *B. Bifurcated Folding Module*

The input layer of conventional object detectors is often a convolutional layer. The convolutional layer not only extracts information in the channel but also models spatial dependencies to extract spatial information. It works well when using the normal RGB image as the input since the semantics in channels are often weak. However, the TAF tensor has very rich temporal information encoded in the channels. Therefore, we design a module called the Bifurcated Folding Module (BFM) to extract semantic information in the temporal dimension of the TAF tensor before feeding it into the detector, which is point-wisely applied at each spatial location.

The overall structure of the BFM is shown in Fig. 6(a). The design of the BFM follows two main concepts. First, instead of being fully connected to the temporal dimension, it will gradually aggregate values at adjacent temporal positions to prevent over-fitting. Second, it will assign greater importance to recent information as it has a greater temporal resolution.

Following the first concept, we design the Folding Layer. The structure of the Folding Layer is shown in Fig. 6(b). The Folding Layer gradually reduces the number of input channels using the $1 \times 1$ depth-wise convolution. It actually models the local connection of adjacent temporal positions. The structure of the Folding Layer and its function is similar to Temporal Convolutional Network (TCN) [34], so we imitate the design of TCN with Weight Normalization and ReLU activation followed.

Following the second concept, we are inspired by the Cross-Stage-Partial-connections (CSP) [35] mechanism. After each temporal aggregation, we make a slicing in the channel dimension to get the channels encoding the temporally latest information and finally connect all sliced channels. This operation also acts as the residual connection, making the module easier to converge during training [36]. We finally fuse the information encoded in output channels with a Multi-Layer Perceptron (MLP). The structure of the MLP is shown in Fig. 6(c).

Some alternative approaches such as Temporal Convolutional Network (TCN) [34], Recurrent Neural Networks (RNNs) [37], [38], and Transformer-based models [39] can be used to extract temporal information from TAF tensors. However, the BFM model has the following advantages. Firstly, BFM is more lightweight and computationally efficient, which makes it more suitable for event data that has a high temporal resolution. Secondly, BFM is more closely combined with TAF event representation since it is designed to model the local connection of adjacent temporal positions rather than long-range temporal dependencies. Moreover, it also explicitly assigns great importance to the temporal latest information in the TAF data since they have a higher temporal resolution.

### C. Agile Event Detector

In the field of generic object detection, YOLO series [19], [22], [32] are featured by the outstanding trade-off between accuracy and efficiency. The YOLO series has the advantage of high speed over other approaches such as Faster R-CNN [40], Mask R-CNN [41], RetinaNet [42], and DETR [43], while maintaining competitive detection accuracy. Hence, we customize an Agile Event Detector (AED) based on the YOLOX model [44], which is more lightweight and yields higher speed compared to the YOLOX baseline. The general structure of the AED model is shown in Fig. 7.

Different from the YOLOX's CSPDarknet [35] backbone, AED utilizes a backbone network adapted from the Darknet21 used in YOLOv3 [45]. Since there is no channel connection operation, Darknet21 runs faster than CSPDarknet. In addition, we increase the channel number of feature maps at an early stage since the event representation tensors encode rich semantics in the channels. The Feature Pyramid Network (FPN) and the detection head are modified from the YOLOX as shown in Fig. 7.

To improve the generalization capability of the lightweight detector, we also propose a simple but effective data augmentation method for event stream representation tensor data:

1) Random Flipping: During training, each event representation tensor $S$ has $p_1$ probability to be flipped horizontally, *i.e.*, $S^*_{c,y,x} = S_{c,y,W-x-1}$.
2) Random Cropping and Resizing: During training, each event representation tensor $\mathrm{S} \in \mathbb{R}^{C\times H\times W}$ has $p_2$ probability to be resampled to $C \times \lfloor \alpha H \rfloor \times \lfloor \alpha W \rfloor$ using nearest neighbor interpolation, where $\alpha \in [1, \infty)$, and then randomly cropped back to $C \times H \times W$, *i.e.* $S^*_{c,y,x} = S_{c,y^*:y^*+H,x^*:x^*+W}$, where $y^* \sim U(0, \lfloor \alpha H \rfloor - H), x^* \sim U(0, \lfloor \alpha W \rfloor - W)$, and $U(a, b)$ is the uniform distribution on $[a, b], a, b \in \mathbb{R}$.

## V. Experimental Results

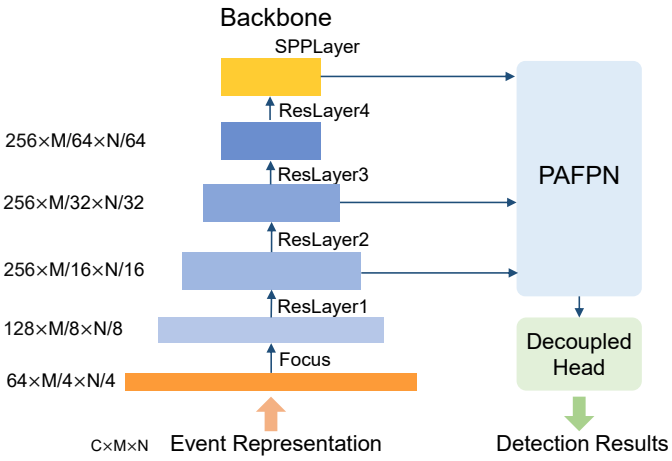



Fig. 7: The structure of the AED model. It contains a backbone network adopted from the Darknet21 [45]. It also contains an FPN and a detection head the same as the implementation in YOLOX [44]. $(C \times M \times N)$ is the size of the input event representation tensor.

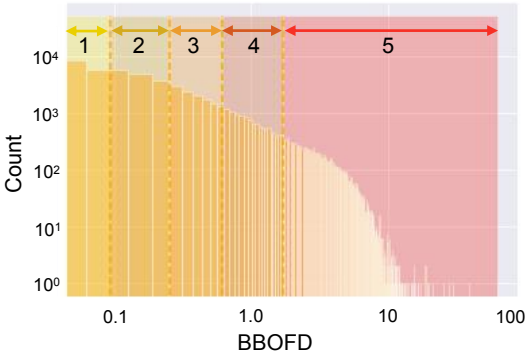
In this section, we will present our experimental settings, including details of the dataset used for the experiment, the selection of hyper-parameters, and the evaluation method. Then we will compare our method with state-of-the-art methods. Finally, we will illustrate the effectiveness of each component of our method through ablation studies.

### A. Experiment Settings

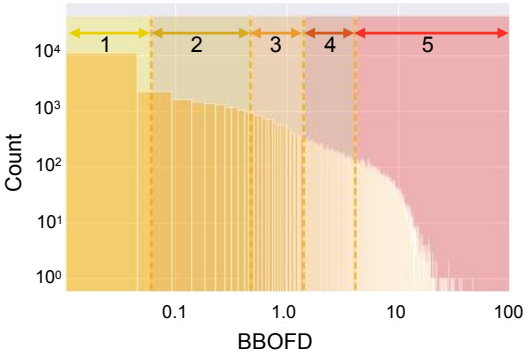
*1) Dataset:* We choose the Prophesee GEN1 Automotive Detection Dataset (GEN1 Dataset) [26] and the Prophesee 1 MEGAPIXEL Automotive Detection Dataset (1 MEGAPIXEL Dataset) [20] to conduct the experiments. The 1 MEGAPIXEL Dataset and the GEN1 Dataset are two typical real-scene event camera object detection datasets. Each video stream is up to 60 seconds, captured with the event camera and stored as the event stream. The timestamps are in microsecond resolution, therefore $T_{max} = 6 \times 10^7 \mu s$.

To improve the efficiency of performance validation, our experiments are conducted on the complete GEN1 Dataset and the downsampled 1 MEGAPIXEL Dataset (1 MEGAPIXEL Dataset (Subset)) by reducing the annotation frequency to 1 Hz while preserving the complete event stream data. This downsampling approach preserves as much diversity of scenes and objects as possible. However, since the number of samples used for training is only 1/60 of the original dataset, the performance of the trained model may be degraded.

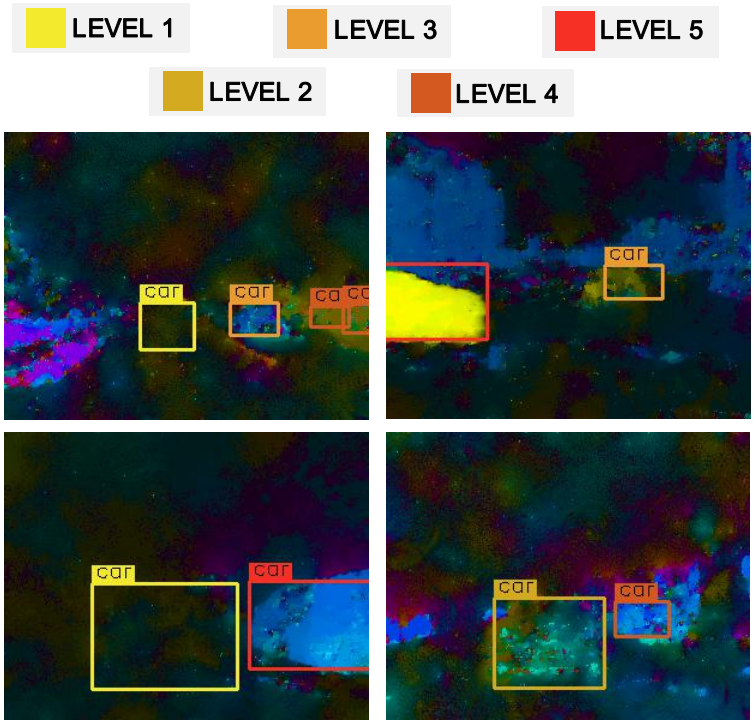
*2) Implementation Details:* Referring to the setting in YOLOX [44], all methods based on YOLOX and AED models are trained using Adam optimizer [46] and cosine learning




(a) Histogram of BBOFD values of all objects and the motion level segmentation on the GEN1 Dataset




(b) Histogram of BBOFD values of all objects and the motion level segmentation on the 1 MEGAPIXEL Dataset (Subset)




(c) Optical flow of objects in different motion level. The bounding box color of an object shows its motion level.

Fig. 8: The visualization of data segmentation and examples of objects under different motion levels.

rate with 5 preheat epochs. In the preheat epochs, the learning rate will grow from 0 to $2.1 \times 10^{-4} \times Batchsize$, and then decrease. The Batch size is taken as 30 on GEN1 Dataset and 16 on the 1 MEGAPIXEL Dataset (Subset). We use the best training model on the validation dataset and apply it to the testing dataset to report the final performance, which is a widely used method in related works [20], [21]. For the data augmentation, we take $p_1 = 0.5$, $p_2 = 0.5$, and $\alpha = 1.5$. For all Event Volume representation tensors, we take $B = 5$, which is commonly used in existing work [20]. For the TAF tensor, we set $\Delta\tau = 10ms$ on both datasets since it approximates the run time of the method. All methods in our work were trained on a server with a single GeForce RTX 3090 GPU and an 8-core Intel(R) Core(TM) i7-7820X CPU @ 3.60GHz. We test all methods on a server with a single Titan Xp GPU and a 16-core Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz.

*3) Evaluation Method:* The performance metrics include accuracy, running time, and the number of parameters. The accuracy evaluation protocol is consistent with the evaluation protocol provided by the 1 MEGAPIXEL Dataset [20]. In the protocol, the Mean Average Precision (mAP) [**?**] is used as the accuracy metric, which is a commonly used metric for object detection models. mAP (IoU@[0.5:0.05:0.95]) is used for evaluation, where the mAP is computed by the mean over a range of IoU thresholds from 0.5 to 0.95 with a step size of 0.05. For algorithms that use the TAF tensor as input, we set the detection period equal to $\Delta\tau$ and the tolerance to $\Delta\tau/2$; for algorithms using other event representation methods, we specify detections at timestamps where the annotations appear.

To evaluate the motion robustness of methods, we measure the motion speed of objects in the dataset by computing optical flow, to classify them into 5 different motion levels. To be specific, we compute the TV-L1 dense optical flow using the method proposed by Nagata et al. [33] for all timestamps where the annotations are present. The difference between adjacent timestamps is set to $50ms$. For timestamp $t_j$ with an annotation $b_j^*$ presents, for a camera with the picture size of $H$ in height and $W$ in width, let the estimated horizontal optical flow be $u_{t_j}(x, y)$ and the vertical optical flow be $v_{t_j}(x, y)$, then we have the optical flow intensity denoted as:

$$\begin{aligned} ||V(x,y)||_{t_j} &= \sqrt{u_{t_j}^2(x,y) + v_{t_j}^2(x,y)} \\ \forall \quad x \in \{0,1,...,W-1\}, y &\in \{0,1,...,H-1\} \end{aligned} \tag{6}$$

Based on the optical flow intensity, we can define a metric called Bounding Box Optical Flow Density (BBOFD) for each annotation $b_j^*$:

$$||V(x,y)||_j = \frac{\sum_{x=x_j}^{x_j+w_j-1}\sum_{y=y_j-h_j+1}^{y_j}||V(x,y)||_{t_j}}{w_j \times h_j} \quad (7)$$

For the object corresponding to the annotation $b_j^*$, BBOFD can be an effective metric for its motion speed relative to the event camera at $t_j$. The larger the $||V(x,y)||_j$, the faster the object. Of all the annotations in each of the two datasets, we set 5 intervals for the BBOFD at each 20% quantile. The intervals can classify the annotations and detection results into 5 classes according to their corresponding objects' speed relative to the camera. We call the 5 classes the 5 motion levels, which are numbered from 1-5, indicating the speed from low to high. We show the visualization of data segmentation and examples of objects under different motion levels in Fig. 8.

To make the metric accurate, we filtered out all overlapping bounding boxes and cropped all bounding boxes into the camera picture size. We then evaluate the detection results under each of the five classes.

### *B. Comparison with the State-of-the-art*

We compare our method with the state-of-the-art methods [20]–[22], [24], [32], [48]. We first compare our overall method with other existing works in terms of accuracy, model inference speed, and model parameters. Then, for the event representation methods, we conduct a separate comparison experiment, using AED as the detector, comparing the accuracy and speed of TAF with other methods.

*1) Comparison with the State-of-the-art methods :* TABLE I shows the results of comparing our method with the state-of-the-art methods [22], [48]. It can be seen that among the methods using feed-forward detectors without the memory mechanism [22]–[24], [32], [48], our method achieves the best performance in terms of accuracy and speed at the same time. Compared with the NGA-events method [32], our method needs only 24.1% of parameters but achieves 9.5 mAP points improvement and 54.1% model inference time reduction on the GEN1 Dataset. Our method also achieves 9.6 mAP points improvement and 37.1% inference time reduction on the 1 MEGAPIXEL Dataset (Subset). ASTMNet [21] is currently the most advanced method on GEN1 Dataset. Comparing our method with ASTMNet, the mAP is 1.3 points lower on GEN1 Dataset, but the model inference time reduces by 66.4%, and the number of model parameters is only 37.4%. It can be seen that our method is competitive among existing methods when jointly considering the accuracy, speed, and model size metrics.

*2) Comparison with the State-of-the-art event representation methods:* Based on the AED with data augmentation, we compare our TAF representation with three state-of-the-art competitors, namely, Event Volume [29], Event Count Image [17], [27], and Surface of Active Events [27], [30]. We set three sets of hyper-parameters for each of the three event representation methods, resulting in different global synchronized time windows and temporal resolutions.

TABLE II and III show the comparison of accuracy between different event representation methods with different parameters at 5 motion levels in the GEN1 Dataset and the 1 MEGAPIXEL Dataset (Subset) respectively. Though with different settings, the accuracy of different methods varies at each motion level, different methods achieve the highest accuracy on motion level 3 in general. On the one hand, as the motion level goes lower, the accuracy reduces, since the events are triggered less frequently and event representation methods cannot aggregate enough information. On the other hand, as the motion level goes higher, the accuracy also degrades, which results from the fact that fast-moving objects generate excessive events, causing motion blur in the event representation. It can also be found that in general the accuracy on the motion level 1 is much lower than that on level 5. This indicates that in the event data object detection task, the accuracy degradation caused by insufficient information is more severe than that caused by motion blur. It can be seen that our TAF method is highly competitive in terms of both accuracy and speed. When using Event Count Image, Surface of Active Events, and Event Volume, longer time windows should be used to improve the detection accuracy for objects with low motion levels. On the contrary, shorter time windows should be used for objects with high motion levels. This demonstrates the trade-off that exists in these event representation methods. By contrast, the improvement in the time window and temporal resolution makes the TAF method achieves high detection accuracy at all motion levels in a hyper-parameter-free manner. Especially when detecting low-motion level objects, using the TAF method can achieve quite significant accuracy improvements.

The visualization in Fig. 9 further demonstrates the robustness of the TAF method. In case (a), the first object on the left has a high motion speed relative to the camera, so when using Event Volume taking $\Delta\tau = 200ms$ and Surface of Active Events taking $\lambda = 1 \times 10^{-6}$, both cannot detect the object due to the motion blur. Although the object can be detected when using Event Count Image under both $N = 50,000$ and $N = 200,000$, the estimation of the height is inaccurate. In case (d), the two objects on the right side have high motion speed relative to the camera. Therefore, also due to the motion blur, the estimation of the size is inaccurate when using Event Volume taking $\Delta\tau = 200ms$, the localization is inaccurate when using Event Count Image taking $N = 200,000$, and the object is not detectable when using Surface of Active Events taking $\lambda = 1 \times 10^{-6}$. On the other hand, in both cases (b)(c)(d), there are objects with low motion speed relative to the camera. It can be seen that when using Event Volume taking $\Delta\tau = 50ms$ and Surface of Active Events taking $\lambda = 1 \times 10^{-5}$, the first object on the left in case (b) and the second object on the left in case (c) are not detected, while the size estimation of the first object on the left in case (d) is not accurate. When using Event Count Image taking $N = 50,000$, the second object from the left in case (c) is not detected and the size of the first object from the left in case (d) is not estimated accurately. In contrast, the TAF method can detect

TABLE I: Performance comparison with the state-of-the-art methods. The bold is the best result in each group of comparisons.

| Method | Event Representation | Detector | Memory | GEN1 Dataset | | 1 MEGAPIXEL Dataset (Subset) | | Params(M) |
|---|---|---|---|---|---|---|---|---|
| | | | | mAP | Inference Time(ms) | mAP | Inference Time(ms) | |
| Chen et al. [22] | Event Count Image [28] | YOLO [47] | - | 0.322 | 21.47 | - | - | 45.3 |
| Jiang et al. [48] | Event Count Image [28] | YOLOv3 [45] | - | 0.326 | 22.34 | 0.207 | 20.81 | 61.5 |
| JDF-events [24] | 2 Polarities Event Count Image [28] | YOLOv3 [45] | - | 0.332 | 22.34 | 0.224 | 20.81 | 61.5 |
| NGA-events [32] | Event Volume [29] | YOLOv3 [45] | - | 0.359 | 26.11 | 0.248 | 21.23 | 61.5 |
| Sparse-conv [23] | Raw Events | YOLO [47] | - | 0.145 | - | - | - | - |
| RED [20] | Event Volume [29] | SSD [49] | ConvLSTM [50] | 0.400 | - | - | - | 24.1 |
| ASTMNet [21] | Raw Events | SSD [49] | Rec-conv [21] | **0.467** | 35.61 | - | - | 39.6 |
| Our baseline | Event Volume [29] | YOLOX [44] | - | 0.350 | 13.19 | 0.213 | 16.16 | **14.4** |
| Ours | Temporal Active Focus | AED | - | 0.454 | **11.98** | **0.344** | **13.36** | 14.8 |

TABLE II: Comparison of accuracy between different event representation methods with different parameters at 5 motion levels on GEN1 Dataset. The bold is the best result in each group of comparisons.

| Method | $\Delta\tau(ms)$ | $N$ | $\lambda$ | Lv1 mAP | Lv2 mAP | Lv3 mAP | Lv4 mAP | Lv5 mAP | Overall mAP | Representation Time(ms) |
|---|---|---|---|---|---|---|---|---|---|---|
| Event Volume [29] | 50 | - | - | 0.172 | 0.414 | 0.455 | 0.451 | 0.397 | 0.426 | 2.60 |
| | 100 | | | 0.198 | 0.428 | 0.453 | 0.441 | 0.382 | 0.424 | 2.80 |
| | 200 | | | 0.218 | 0.434 | 0.459 | 0.448 | 0.374 | 0.422 | 3.27 |
| Event Count Image [17], [27] | - | $5 \times 10^4$ | - | 0.204 | 0.371 | 0.408 | 0.411 | 0.362 | 0.386 | **0.90** |
| | | $1 \times 10^5$ | | 0.211 | 0.377 | 0.405 | 0.395 | 0.343 | 0.376 | 1.01 |
| | | $2 \times 10^5$ | | 0.228 | 0.388 | 0.394 | 0.369 | 0.312 | 0.368 | 1.19 |
| Surface of Active Events [27], [30] | - | - | $1 \times 10^{-5}$ | 0.199 | 0.403 | 0.425 | 0.421 | 0.367 | 0.400 | 0.99 |
| | | | $2.5 \times 10^{-6}$ | 0.216 | 0.429 | 0.431 | 0.403 | 0.353 | 0.404 | 1.01 |
| | | | $1 \times 10^{-6}$ | 0.228 | 0.420 | 0.428 | 0.400 | 0.349 | 0.403 | 1.06 |
| Temporal Active Focus | 10 | - | - | **0.282** | **0.461** | **0.476** | **0.459** | **0.407** | **0.454** | 1.43 |

TABLE III: Comparison of accuracy between different event representation methods with different parameters at 5 motion levels on 1 MEGAPIXEL Dataset (Subset). The bold is the best result in each group of comparisons.

| Method | $\Delta\tau(ms)$ | $N$ | $\lambda$ | Lv1 mAP | Lv2 mAP | Lv3 mAP | Lv4 mAP | Lv5 mAP | Overall mAP | Representation Time(ms) |
|---|---|---|---|---|---|---|---|---|---|---|
| Event Volume [29] | 50 | - | - | 0.038 | 0.219 | 0.224 | **0.318** | **0.314** | 0.299 | 7.80 |
| | 100 | | | 0.035 | 0.232 | 0.325 | 0.315 | 0.301 | 0.305 | 11.14 |
| | 200 | | | 0.062 | 0.239 | 0.309 | 0.281 | 0.276 | 0.290 | 14.20 |
| Event Count Image [17], [27] | - | $4 \times 10^5$ | - | 0.111 | 0.209 | 0.269 | 0.257 | 0.271 | 0.276 | **1.31** |
| | | $8 \times 10^5$ | | 0.105 | 0.212 | 0.270 | 0.253 | 0.254 | 0.268 | 1.34 |
| | | $1.2 \times 10^6$ | | 0.131 | 0.233 | 0.258 | 0.240 | 0.260 | 0.278 | 1.49 |
| Surface of Active Events [27], [30] | - | - | $1 \times 10^{-5}$ | 0.043 | 0.231 | 0.315 | 0.287 | 0.286 | 0.294 | 3.71 |
| | | | $2.5 \times 10^{-6}$ | 0.090 | 0.234 | 0.297 | 0.274 | 0.288 | 0.296 | 3.20 |
| | | | $1 \times 10^{-6}$ | 0.143 | 0.262 | 0.300 | 0.272 | 0.296 | 0.303 | 3.32 |
| Temporal Active Focus | 10 | - | - | **0.262** | **0.312** | **0.339** | 0.308 | **0.314** | **0.344** | 2.83 |

TABLE IV: The ablation of each component in our method. The bold is the best result in each group of comparisons.

| Detector | Data Augmentation | Event Representation | GEN1 Dataset | | 1 MEGAPIXEL Dataset (Subset) | | Params(M) |
|---|---|---|---|---|---|---|---|
| | | | mAP | Runtime(ms) | mAP | Runtime(ms) | |
| YOLOX [44] | - | Event Volume [29] | 0.350 | 15.79 | 0.213 | 23.96 | **14.4** |
| YOLOX [44] | ✓ | Event Volume [29] | 0.410 | 15.79 | 0.269 | 23.96 | **14.4** |
| AED | ✓ | Event Volume [29] | 0.426 | 14.18 | 0.299 | 21.35 | 14.8 |
| AED | ✓ | TAF | **0.454** | **13.41** | **0.344** | **16.19** | 14.8 |

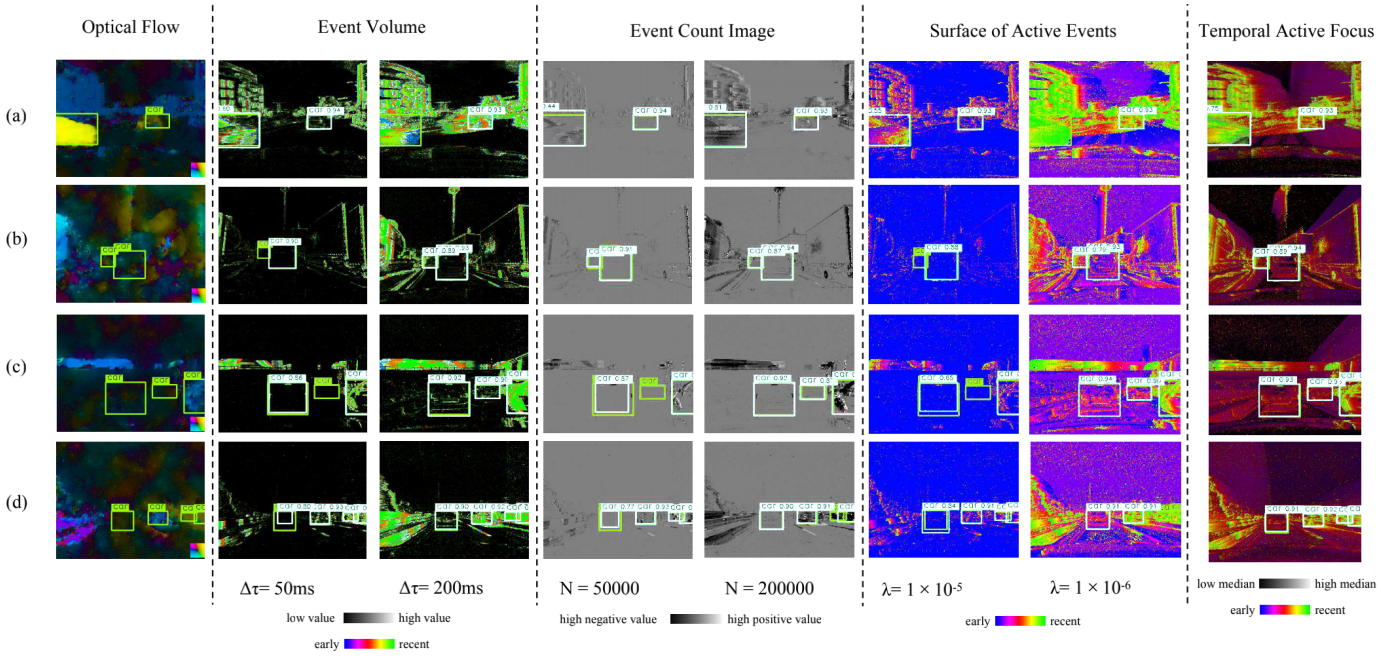



Fig. 9: Qualitative analysis results. The green bounding boxes indicate the annotations, while the white bounding boxes indicate the detection results.

TABLE V: The effects of inserting TAF to SOTA methods. The bold is the best result in each group of comparisons.

| Detector | Event Representation | GEN1 Dataset | | 1 MEGAPIXEL Dataset (Subset) | | Params(M) |
|---|---|---|---|---|---|---|
| | | mAP | Runtime(ms) | mAP | Runtime(ms) | |
| YOLOv3 [45] | Event Volume [29] | 0.359 | 28.71 | 0.248 | 29.03 | **61.5** |
| | TAF | **0.381** | **27.54** | **0.278** | **24.06** | 61.6 |
| YOLOX [44] | Event Volume [29] | 0.410 | 15.79 | 0.269 | 23.96 | **14.4** |
| | TAF | **0.436** | **14.62** | **0.314** | **18.99** | 14.4 |
| AED | Event Volume [29] | 0.426 | 14.18 | 0.299 | 21.35 | **14.8** |
| | TAF | **0.454** | **13.41** | **0.344** | **16.19** | 14.8 |

all the objects mentioned above while estimating the size and location accurately.

### C. Ablation Study

In the ablation experiments, we focus on the effectiveness of the different components of our method and factors that affect the performance when using the TAF representation method.

*1) Components:* The ablation of components is shown in TABLE IV, where the runtime is the sum of the representation time and the model inference time. Experiments show that our proposed data augmentation scheme can improve the accuracy of the baseline detector by a large margin on both datasets. With a slight increase in the number of parameters compared to YOLOX, AED notably improves accuracy and significantly reduces runtime on both datasets. Moreover, using the TAF method instead of the Event Volume will result in a significant improvement in accuracy and a reduction of the running time on both datasets. The reduction of the running time is especially significant on the 1 MEGAPIXEL Dataset (Subset). Our final method achieves 74.6 FPS on the Prophesee GEN1 Automotive Detection Dataset (GEN1 Dataset) [26] and 61.8 FPS on the Prophesee 1 MEGAPIXEL Automotive Detection Dataset [20], which satisfies the real-time processing requirements.

We also evaluate the effectiveness of inserting TAF into various SOTA event data object detection methods. As shown in TABLE V, our experimental results show that TAF significantly improves the detection accuracy of these methods while maintaining a high inference speed. The results show the generalizability of our proposed TAF method.

*2) Temporal Active Focus:* TABLE VI and VII show the performance of the TAF under different settings. Overall, using BFM for feature pre-extracting will result in remarkable accuracy improvement at all five motion levels on both datasets. On the 1 MEGAPIXEL Dataset (Subset), we can see a higher performance improvement for detecting objects that are slow relative to the camera. However, the running speed is slightly reduced because feature extraction needs to be performed point-wisely in space. On GEN1 Dataset, the queue depth $K$ value has little impact on accuracy, and we can see that $K = 4$ is a better hyper-parameter on GEN1 Dataset. On the 1 MEGAPIXEL Dataset (Subset), when $K = 8$,

TABLE VI: The performance of the TAF under different settings on the GEN1 Dataset. The bold is the best result in each group of comparisons.

| $K$ | BFM | Lv1 mAP | Lv2 mAP | Lv3 mAP | Lv4 mAP | Lv5 mAP | Overall mAP | Representation Time(ms) | Inference Time(ms) |
|---|---|---|---|---|---|---|---|---|---|
| 4 | - | 0.272 | 0.438 | 0.456 | 0.444 | 0.396 | 0.444 | **1.43** | **11.46** |
| 4 | ✓ | 0.282 | **0.461** | 0.476 | **0.459** | **0.407** | **0.454** | **1.43** | 11.98 |
| 8 | - | 0.269 | 0.439 | 0.456 | 0.449 | 0.398 | 0.445 | 1.76 | 11.67 |
| 8 | ✓ | **0.288** | 0.451 | **0.478** | 0.457 | 0.402 | 0.451 | 1.76 | 12.26 |

TABLE VII: The performance of the TAF under different settings on 1 MEGAPIXEL Dataset (Subset). The bold is the best result in each group of comparisons.

| $K$ | BFM | Lv1 mAP | Lv2 mAP | Lv3 mAP | Lv4 mAP | Lv5 mAP | Overall mAP | Representation Time(ms) | Inference Time(ms) |
|---|---|---|---|---|---|---|---|---|---|
| 4 | - | 0.230 | 0.284 | 0.313 | 0.293 | 0.301 | 0.326 | **1.76** | **11.67** |
| 4 | ✓ | 0.222 | 0.270 | 0.313 | 0.291 | 0.300 | 0.333 | **1.76** | 12.84 |
| 8 | - | 0.258 | 0.296 | 0.315 | 0.294 | 0.300 | 0.323 | 2.83 | 12.38 |
| 8 | ✓ | **0.262** | **0.312** | **0.339** | **0.308** | **0.314** | **0.344** | 2.83 | 13.36 |

the accuracy is slightly lower without BFM. However, with BFM, the accuracy is much higher. This further illustrates the effectiveness of the BFM module: the 1 MEGAPIXEL Dataset (Subset) has a higher resolution, and the camera used is more sensitive to changes in illumination, requiring larger $K$ values to aggregate events over a larger time range to fully retain event information. However, larger $K$ values bring richer semantics to the temporal dimension. The traditional convolutional input layer is difficult to extract the information. On the other hand, BFM can extract the rich semantics as much as possible, thus fully exploiting the effect of the TAF method.

## VI. Conclusions

In this paper, we present a motion robust and high-speed detection pipeline for event-based object detection, which takes the different velocities of objects into account and further reduces the computational burden compared with previous event-based object detectors. Specifically, we introduce the Temporal Active Focus (TAF) event representation, the Bifurcated Folding Module (BFM), the Agile Event Detector (AED), and a simple yet effective data augmentation strategy. The TAF leverages the spatial-temporal asynchronous event data and builds event tensors robust to object motions, the BFM extracts rich temporal information at the input layer, and the AED is faster and more accurate than the baseline YOLOX. Extensive experiments on two typical event-based object detection datasets show that our detection pipeline achieves leading accuracy compared with the state-of-the-art event-based object detector. In addition, our method has a far lower number of parameters and much higher running speed while achieving competitive accuracy and high motion robustness.

## Acknowledgement

We would like to thank Mr. Etienne Perot, Senior Machine Learning Engineer from Prophesee, for providing the implementation details of the methods in [20]. We would like to thank Mr. Jianing Li, Ph.D. Candidate from Multimedia Learning Group, National Engineering Laboratory for Video Technology, Peking University for providing details about the experimental conditions and the size of the neural network models in [21].

## References

[1] Y. Cai, T. Luan, H. Gao, H. Wang, L. Chen, Y. Li, M. A. Sotelo, and Z. Li, "Yolov4-5d: An effective and efficient object detector for autonomous driving," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.

[2] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341–1360, 2020.

[3] S. Cattini, D. Cassanelli, G. Di Loro, L. Di Cecilia, L. Ferrari, and L. Rovati, "Analysis, quantification, and discussion of the approximations introduced by pulsed 3-d lidars," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.

[4] W. Jang, M. Park, and E. Kim, "Real-time driving scene understanding via efficient 3-d lidar processing," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–14, 2022.

[5] X. Li, Y. Zhou, and B. Hua, "Study of a multi-beam lidar perception assessment model for real-time autonomous driving," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–15, 2021.

[6] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 1907–1915.

[7] Z. Zhang, X. Wang, D. Huang, X. Fang, M. Zhou, and Y. Zhang, "Mrpt: Millimeter-wave radar-based pedestrian trajectory tracking for autonomous urban driving," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–17, 2021.

[8] S. Pagad, D. Agarwal, S. Narayanan, K. Rangan, H. Kim, and G. Yalla, "Robust method for removing dynamic objects from point clouds," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 765–10 771.

[9] S. Shirmohammadi and A. Ferrero, "Camera as the instrument: the rising trend of vision based measurement," *IEEE Instrumentation & Measurement Magazine*, vol. 17, no. 3, pp. 41–47, 2014.

[10] T. Serrano-Gotarredona and B. Linares-Barranco, "A 128 $\times$128 1.5% contrast sensitivity 0.9% fpn 3 $\mu$s latency 4 mw asynchronous frame-free dynamic vision sensor using transimpedance preamplifiers," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 3, pp. 827–838, 2013.

[11] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128× 128 120 db 15 $\mu$s latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[12] T. Finateu, A. Niwa, D. Matolin, K. Tsuchimoto, A. Mascheroni, E. Reynaud, P. Mostafalu, F. Brady, L. Chotard, F. LeGoff *et al.*, "5.10 a 1280× 720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86 $\mu$m pixels, 1.066 geps readout, programmable event-rate controller and compressive data-formatting pipeline," in *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2020, pp. 112–114.

[13] H. Cao, G. Chen, Z. Li, Y. Hu, and A. Knoll, "Neurograsp: multimodal neural network with euler region regression for neuromorphic vision-based grasp pose estimation," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.

[14] C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, and T. Delbruck, "Retinomorphic event-based vision sensors: bioinspired cameras with spiking output," *Proceedings of the IEEE*, vol. 102, no. 10, pp. 1470–1484, 2014.

[15] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis *et al.*, "Event-based vision: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 1, pp. 154–180, 2020.

[16] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, and A. Knoll, "Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 34–49, 2020.

[17] A. I. Maqueda, A. Loquercio, G. Gallego, N. García, and D. Scaramuzza, "Event-based vision meets deep learning on steering prediction for self-driving cars," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5419–5427.

[18] J. Binas, D. Neil, S.-C. Liu, and T. Delbruck, "Ddd17: End-to-end davis driving dataset," *arXiv preprint arXiv:1711.01458*, 2017.

[19] M. Cannici, M. Ciccone, A. Romanoni, and M. Matteucci, "Asynchronous convolutional networks for object detection in neuromorphic cameras," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 1656–1665.

[20] E. Perot, P. de Tournemire, D. Nitti, J. Masci, and A. Sironi, "Learning to detect objects with a 1 megapixel event camera," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 16 639–16 652.

[21] 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 Transactions on Image Processing*, vol. 31, pp. 2975–2987, 2022.

[22] N. F. Chen, "Pseudo-labels for supervised learning on dynamic vision sensor data, applied to object detection under ego-motion," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 644–653.

[23] N. Messikommer, D. Gehrig, A. Loquercio, and D. Scaramuzza, "Event-based asynchronous sparse convolutional networks," in *European Conference on Computer Vision*. Springer, 2020, pp. 415–431.

[24] J. Li, S. Dong, Z. Yu, Y. Tian, and T. Huang, "Event-based vision enhanced: A joint detection framework in autonomous driving," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2019, pp. 1396–1401.

[25] D. Gehrig, A. Loquercio, K. G. Derpanis, and D. Scaramuzza, "End-to-end learning of representations for asynchronous event-based data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5633–5643.

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

[27] A. Z. Zhu and L. Yuan, "Ev-flownet: Self-supervised optical flow estimation for event-based cameras," in *Robotics: Science and Systems*, 2018.

[28] H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization," in *British Machine Vision Conference (BMVC)*, 2017.

[29] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 989–997.

[30] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-based visual flow," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 2, pp. 407–417, 2013.

[31] R. Baldwin, R. Liu, M. M. Almatrafi, V. K. Asari, and K. Hirakawa, "Time-ordered recent event (tore) volumes for event cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 2519–2532, 2023.

[32] Y. Hu, T. Delbruck, and S.-C. Liu, "Learning to exploit multiple vision modalities by using grafted networks," in *European Conference on Computer Vision*. Springer, 2020, pp. 85–101.

[33] J. Nagata, Y. Sekikawa, and Y. Aoki, "Optical flow estimation by matching time surfa ce with event-based cameras," *Sensors*, vol. 21, no. 4, p. 1150, 2021.

[34] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 156–165.

[35] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[37] M. Jordan, "Serial order: A parallel distributed processing approach," in *Advances in psychology*. Elsevier, 1997, vol. 121, pp. 471–495.

[38] J. S. S. Hochreiter, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[40] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[41] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[42] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[43] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[44] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.

[45] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *The 3rd International Conference for Learning Representations*, 2015.

[47] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

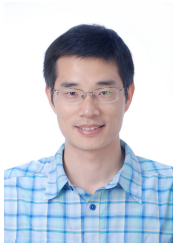
[48] Z. Jiang, P. Xia, K. Huang, W. Stechele, G. Chen, Z. Bing, and A. Knoll, "Mixed frame-/event-driven fast pedestrian detection," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8332–8338.

[49] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[50] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, vol. 28, 2015, pp. 802–810.

**Bingde Liu** received his B.S. degree in Communication Engineering from Wuhan University, Wuhan, China, in 2022, his MSc. degree in Financial Technology with Data Science from University of Bristol, Bristol, U.K., in 2023, he is currently pursuing his Ph.D. degree in Industrial Engineering and Economics at Tokyo Institution of Technology, Tokyo, Japan. His research focuses on the application of artificial intelligence.

**Chang Xu** received his B.S. degree in electronic information engineering from Wuhan University, Wuhan, China, in 2021, he is currently pursuing his M.S. degree in communication and information system at Wuhan University, Wuhan, China. His research focuses on tiny object detection, oriented object detection, and label noise.



**Wen Yang** (Senior Member, IEEE) received his B.S. degree in Electronic Apparatus and Surveying Technology, his M.S. degree in Computer Application Technology, and his Ph.D. degree in Communication and Information System, all from Wuhan University, Wuhan, China, in 1998, 2001, and 2004, respectively. In 2008 and 2009, he worked as a Visiting Scholar with the Apprentissage et Interfaces (AI) Team at the Laboratoire Jean Kuntzmann in Grenoble, France. Following that, he served as a Post-Doctoral Researcher with the State Key Laboratory of Information Engineering, Surveying, Mapping, and Remote Sensing, also at Wuhan University, from 2010 to 2013. Since then, he has held the position of Full Professor at the School of Electronic Information, Wuhan University. His research interests include object detection and recognition, multisensor information fusion, and remote sensing image interpretation.

**Huai Yu** (Member, IEEE) received his B.S. and Ph.D. degrees in communication and information systems from Wuhan University, China, in 2015 and 2020, respectively. From 2018 to 2020 and 2020 to 2021, he has been a visiting scholar and postdoctoral fellow at the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. He is currently working as a research associate professor at the School of Electronic Information, Wuhan University. His research involves multi-modal visual feature detection and matching, structure from motion, and SLAM.

**Lei Yu** received his B.S. and Ph.D. degrees in signal processing from Wuhan University, Wuhan, China, in 2006 and 2012, respectively. From 2013 to 2014, he has been a Postdoc Researcher with the VisAGeS Group at the Institut National de Recherche en Informatique et en Automatique (INRIA) for one and a half years. He is currently working as an associate professor at the School of Electronics and Information, Wuhan University, Wuhan, China. From 2016 to 2017, he was also a Visiting Professor at Duke University for one year. He has been working as a guest professor in the École Nationale Supérieure de l'Électronique et de ses Applications (ENSEA), Cergy, France, for one month in 2018. His research interests include neuromorphic vision and computation.