Motion and Appearance Decoupling Representation for Event Cameras

Nuo Chen, Boyang Li, Yingqian Wang, *Member, IEEE*, Xinyi Ying, Longguang Wang, Chushu Zhang, Yulan Guo, *Senior Member, IEEE*, Miao Li, and Wei An

***Abstract*—Event cameras, with high temporal resolution and high dynamic range, have shown great potential under extreme scenarios such as high-speed movement and low illumination. However, previous event representation methods typically aggregate event data into a single dense tensor, often overlooking the dynamic changes of events within a given time unit. This limitation can introduce historical artifacts and semantic inconsistencies, ultimately degrading model performance. Inspired by human visual prior, we propose a *motion and appearance decoupling* (MAD) event representation to disentangle the mixed spatial-temporal event tensor into two independent branches. This bio-inspired design helps the network extract discriminative temporal (i.e., motion) and spatial (i.e., appearance) information, thus reducing the network's learning burden toward complex high-level interpretation tasks. In our method, the event motion guided attention module (EMGA) is designed to achieve temporal and spatial feature interaction and fusion sequentially. Based on EMGA, three specially designed decoder heads are proposed for several representative event-based tasks (i.e., object detection, semantic segmentation, and human pose estimation). Experimental results demonstrate that our method achieves state-of-the-art performance on the above three tasks, which reveals that our method is an easy-to-implement replacement for currently event-based methods. Our code is available at: https://github.com/ChenYichen9527/MAD-representation**

***Index Terms*—Event camera, event representation, object detection, semantic segmentation, human pose estimation.**

I. Introduction

EVENT cameras, with high temporal resolution, wide dynamic range, reduced redundancy, and low power consumption, have exhibited significant potential in autonomous driving [1], [2], [3], robotics [4], [5], and security surveillance [6], [7]. Different from traditional frame-based cameras that capture detailed textures of the entire scene at a predetermined frame rate, pixels in event cameras operate asynchronously, producing the location, polarity, and timestamp of every illumination change. This distinct output mechanism results in storing event data in an asynchronous array format, which differs from the conventional dense pixel value matrix acquired at a fixed rate. Due to the above differences, traditional image frame-based neural networks are unsuitable for event-based data. How to properly represent event data becomes a crucial problem.

Sparse representation [8], [9], [10] and dense representation [11], [12], [13] are widely applied event representation approaches. Typical sparse representation-based methods, including spiking neural networks (SNN) based methods [14], [15], [16], point cloud-based methods [9], and graph-based methods [10], [17], directly adopt original events as input. Due to the sparse and unstructured distribution of event data, these methods either have specialized hardware requirements (i.e., SNN) or introduce high computational costs (i.e., point cloud and graph-based methods). There is still a performance gap between these and state-of-the-art methods.

To achieve better performance, dense representation-based methods try to convert events into a dense intermediate representation that is compatible with frame-based processing systems. For example, the event accumulation-based methods [11], [18] and bilinear sampling-based methods [19] reshape the event data as an image sequence by stacking the original input along the temporal dimension with different time intervals. However, these dense representation-based methods simply mix the temporal and spatial information into a dense tensor but ignore the dynamic changes of events within the unit time. This limitation introduces historical artifacts and semantic inconsistencies between event input and ground truth, thus limiting further performance improvements, as shown in Fig. 1(c1).

Inspired by the human visual prior, many representative video image interpretation methods [23], [24], [25] independently treat temporal (i.e., motion) and spatial (i.e., appearance) features. This manner helps the network extract discriminative temporal and spatial features, thus introducing significant performance improvements. Motivated by these methods, firstly, we propose a motion and appearance decoupling (MAD) representation method for event cameras, which decouples the asynchronous event stream into independent motion and appearance information, as shown in Fig 1. MAD first estimates the motion of each event and then aligns them to

Received 15 April 2025; revised 19 June 2025 and 5 August 2025; accepted 17 August 2025. Date of publication 15 September 2025; date of current version 19 September 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62401590. The associate editor coordinating the review of this article and approving it for publication was Dr. Tiesong Zhao. *(Corresponding authors: Wei An; Miao Li.)*

Nuo Chen, Boyang Li, Yingqian Wang, Xinyi Ying, Chushu Zhang, Miao Li, and Wei An are with the College of Electronic Science and Technology, National University of Defense Technology (NUDT), Changsha 410073, China (e-mail: chennuo21@nudt.edu.cn; liboyang20@nudt.edu.cn; wangyingqian16@nudt.edu.cn; yingxinyi18@nudt.edu.cn; zhangchushu21@nudt.edu.cn; lm8866@nudt.edu.cn; anwe@nudt.edu.cn).

Longguang Wang is with the Aviation University of Air Force, Changchun 130012, China (e-mail: wanglongguangl5@nudt.edu.cn).

Yulan Guo is with the School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen Campus, Shenzhen 518107, China (e-mail: yulan.guo@nudt.edu.cn).

Digital Object Identifier 10.1109/TIP.2025.3607632

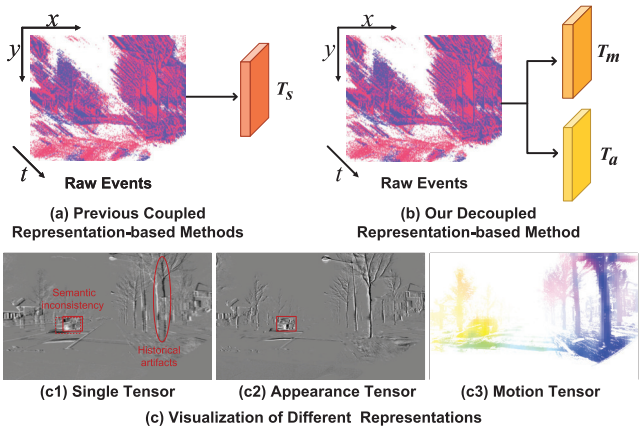


Fig. 1. Comparison between our method and previous methods. Previous methods couple the spatial-temporal information of events into a single tensor T_s, resulting in historical artifacts and semantic inconsistencies between event input (shown as the dashed bounding box) and ground truth (shown as the solid bounding box). Our method decouples the spatial-temporal information and represents raw events as an appearance tensor T_a with clear structures, and a motion tensor T_m with rich movement cues.

a unified timestamp. This eliminates temporal inconsistencies among events, reducing semantic ambiguity during mapping. Additionally, the estimated optical flow serves as auxiliary information, enriching the representation. For example, pixels with consistent motion likely belong to the same object. Then, an event motion-guided attention (EMGA) module is designed to achieve dual-branch feature interaction by using motion features to enhance appearance features. Finally, we design three decoder heads to adopt three typical high-level tasks, including object detection (i.e., FPN-based decoder), semantic segmentation (i.e., Unet-based decoder), and pose estimation (i.e., Dual stream-based decoder). Experimental results (as shown in Fig. 2) show that our method achieves state-of-the-art performance on two event-based object detection datasets (i.e., Gen1 [20] and 1Mpx [26]), two event-based semantic segmentation datasets (i.e., DDD17 [3] and DSEC-Semantic [21]), and an event-based human pose estimation dataset (i.e., DHP19 [22]). These results reveal that our method is an easy-to-implement replacement for current event-based methods. The main contributions can be summarized as follows:

1) We propose a novel event representation method, namely MAD, which decouples the spatial-temporal information of events into two independent tensors. More discriminative motion and appearance features can be extracted in this way.
2) Based on the decoupled tensors, we propose an EMGA module to exploit the motion features to enhance the appearance features for achieving better feature interaction and fusion.
3) We conduct comprehensive experiments to verify the effectiveness of our proposed approach which consistently outperforms the state-of-the-art methods on a wide range of challenging tasks, including object detection, semantic segmentation, and human pose recognition.

II. Related Work

Due to the asynchronous sparsity of event data, its processing steps differ significantly from conventional frame-based computer vision techniques. The first step in event-based computer vision is event representation, where raw DVS data is converted into a specific format compatible with subsequent processing steps (e.g., deep neural networks). These steps include tasks like object detection, semantic segmentation, and human pose estimation.

Existing methods for representing event data can be divided into two categories. The first category focuses on directly processing event data. These methods treat event data as unstructured data, including the direct handling of raw events via SNN [8], [14], [15], [16], [36], [37], the interpretation of events as spatial-temporal point clouds for point cloud-based methods [9], and the modeling of spatial-temporal dynamics through Graph Neural Networks (GNNs) [10], [17], [38]. However, these methods are limited by their dependence on specialized hardware or by performance inadequacies.

The second category of methods, which is the subject of this paper, involves converting event data into an "intermediary representation" that is compatible with machine learning techniques in a synchronized manner. To leverage the rapid advancements of CNNs and their recent success in computer vision, event data needs to be transformed into a proxy 2D image-like or 3D video frame-like representation we call "proxy frames". This category of methods offers better performance, as it can accommodate frame-based computer vision techniques.

Table I summarizes the comparison between event-based "intermediary representations" and their design choices. The representation called "Event Frame" [27] generates $H \times W$ images by summing the polarity of events within a given time window. This approach preserves the spatial distribution of events but ignores polarity and temporal information. A similar approach, called "Event Count Image" [2], counts events within a time window separately for each polarity, resulting in two $H \times W$ proxy frames. This method retains polarity information but loses temporal information.

To capture the spatial-temporal evolution of events, some methods [28], [29], [30] attempt to incorporate temporal information into the event representation. Depending on how the temporal information is encoded, these methods can be divided into two categories. The first category encodes temporal information as pixel values. For instance, Surface of Active Events (SEA) [28] encodes the timestamp of the most recent event as the pixel value of an image, prioritizing the timing of events over edge magnitude. However, this may not be ideal for imaging in complex environments (such as low-light conditions), as the timing of events can be disrupted by noise. Averaged Time Surfaces (ATS) [29] improves upon this by encoding the average timestamp of events within a pixel neighborhood window as the image pixel value. This reduces the impact of isolated noise, however, averaging may lead to temporal confusion in the event representation. The second category encodes temporal information into the channel dimension, generating an "intermediary representation" similar

TABLE I
COMPARISON OF FRAME-BASED EVENT REPRESENTATION METHODS USED IN PRIOR WORK ON EVENT-BASED DEEP LEARNING

Event Representation	Dimensions	Description	Characteristics
Event Frame [27]	$H\times W$	Image of event polarities	Discards temporal and polarity information
Event Count [2]	$2\times H\times W$	Image of event counts	Discards temporal information
SAE [28]	$2\times H\times W$	Image of most recent timestamp	Discards all prior timestamps
ATS [29]	$2\times H\times W$	Image of average timestamp for window	Compress temporal information
Voxel Grid [30]	$B\times H\times W$	Voxel grid summing event polarities	Discards polarity information
MES [31]	$N\times H\times W$	Stacking of different numbers of events	Discards the temporal information within stacking time
EST [32]	$2\times B\times H\times W$	4D grid of convolutions	Temporally quantizes information into B bins
TORE [33]	$2\times K\times H\times W$	4D grid of last K timestamps	Retains information for last K events
Matrix-LSTM [34]	$C\times H\times W$	3D grid of LSTM	Temporally quantizes information into C channels
Group Token [35]	$(H/P\cdot W/P)\times(G\cdot C)$	Group events based on timestamp and polarity	Temporally quantizes information into token
MAD (Ours)	$2\times 2\times H\times W$	Images of clear appearance and event motion	Retains all spatial and temporal information

H and W denote representation height and width, respectively. This table is expanded from [33] to include new methods.

to a 3D video. Voxel grid [30] uses bilinear interpolation to map events to the nearest time window, creating a dense tensor of size $B \times H \times W$, where each $H \times W$ tensor represents spatial information within a $1/B$ time window. Mixed-density Event Stacking (MES) [31] divides events into N multiple overlapping windows that halve the number of events at each stage. This generates a dense tensor of size $N \times H \times W$, where each $H \times W$ tensor represents spatial information of different numbers of events. This is more robust in dynamic scenes with varying speeds of movement.

With the development of deep learning technology, the research of event representation has gradually shifted towards data-driven models, where the mapping of events to "intermediary representation" is automatically learned from data. EST [32] converts asynchronous events into a grid-based representation through kernel convolution, quantization, and projection. This representation enables end-to-end learning from raw event data to task loss. TORE [33] aggregates events into queues, but its computation speed is slow, and its performance is similar to existing voxel grids. Matrix-LSTM [34] applies Long Short-Term Memory networks (LSTM) [39] to 2D event streams, accumulating pixel information over time and constructing dense event tensors. Group Token [35] groups asynchronous events based on their timestamps and polarity, tokenizing events to fit the existing Transformer architecture. These learning-based methods have achieved better performance as compared to hand-crafted feature design methods. However, due to the semantic ambiguity caused by the coupling of spatial-temporal information in a few proxy frames, these representation methods require complex feature extraction networks in subsequent processing steps to achieve better performance. Recently, EvRep [40] generated event representations based on spatial-temporal statistical features to better preserve the inherent temporal patterns of event streams. LOPET [41] proposed a hybrid event-clustering strategy to address the issue of sharp fluctuations in the number of events caused by the rapid or slow movement of objects. In addition, several event representations specifically designed for image deblurring have been proposed [42], [43]. These methods leverage high-frequency event streams to model blur features, thereby achieving high robustness of the network to dynamic scenes. Although these methods have achieved promising performance, they are currently limited to single tasks such as image deblurring and classification. In this work, we propose a spatio-temporal decoupling representation method that transforms event data into two distinct modal tensors (i.e., a motion tensor and an appearance tensor). Our method not only enables the network to more easily learn discriminative features but also exhibits high generalization capability, making it applicable to various downstream tasks.

III. METHODOLOY

In this section, we first introduce how to decouple motion and appearance information from the event stream. Then, we present a fusion module for motion and appearance features. Finally, we introduce three specially designed decoder heads for object detection, semantic segmentation, and human pose estimation, respectively.

A. Decoupling: MAD Representation

Given an event stream E, we first split it into fixed-duration temporal bins $E = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n\}$ as basic processing units. In this way, the input event data ε_i can be described as:

$$\varepsilon_i = \left\{ e_k = (x_k, y_k, t_k, p_k) \,|\, T^i - \Delta T < t_k \leqslant T^i \right\}, \tag{1}$$

where ΔT is the time window, and T^i is the end timestamps of the i-th temporal bin, which we set as the moment when the label is generated. Then, we categorize the events ε_i into positive and negative polarity groups: $\varepsilon_i = \varepsilon_i^+ + \varepsilon_i^-$, where $\varepsilon_i^\pm = \{e_k = (x_k, y_k, t_k, p_k) \,|\, p_k = \pm 1\}$. Finally, the decoupling process involves the following two steps: 1) Motion estimation from the event data ε_i to generate the optical flow tensor $T_m \in \mathbb{R}^{H\times W\times 2}$, where H and W represent camera resolution, and 2 represents the two polarities of the events; 2) Align all events to the unified reference time T_{ref} to reconstruct a clear appearance tensor $T_a \in \mathbb{R}^{H\times W\times 2}$.

1) Motion Tensor: Optical flow refers to the instantaneous velocity of pixel motion observed on the imaging plane of spatial moving objects. Moreover, we consider the motion of events to remain constant within a short time interval (ideally infinitesimal). The motion of events can be represented as an

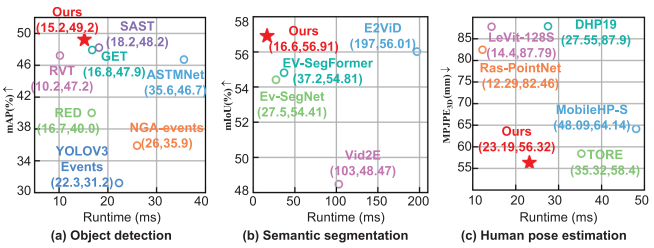


Fig. 2. Comparison of MAD with other methods on three different task datasets (i.e., GEN1 Automotive Dataset [20] for object detection, DSEC-Semantic Dataset [21] for semantic segmentation, and DHP19 dataset [22] for human pose estimation). Our proposed method is highly efficient and effective compared with existing state-of-the-art methods on all these three tasks.

optical flow map $T_m \in \mathbb{R}^{H\times W\times 2}$, where H and W denote the resolution of the camera, and the two channels represent the velocities in the x and y directions, respectively. In this paper, we employ EV-FlowNet [19] to achieve event-based optical flow estimation in an unsupervised manner. More quantitative results can be found in Section IV-C.3.

2) Appearance Tensor: We use optical flow to align the event stream to a reference time t_{ref}, and convert it into an image-like representation with clear contours and edges, thus separating the appearance information from the event stream. Based on the assumption of optical flow consistency, the trajectories followed by events on the image plane are locally straight, approximated by translations: $\mathbf{x}(t) = \mathbf{x}(0) + \mathbf{v}t$, where $\mathbf{x} = (x, y)^T$ and $\mathbf{v}$ is the velocity (i.e., the optical flow) of the event. More specifically, given a set of events $\varepsilon = \{e_k = (x_k, y_k, t_k, p_k)\}_{k=1}^N$, a geometric transformation is performed utilizing optical flow $\mathbf{v}$, producing the aligned events $\varepsilon'_{t_{ref}} = \left\{e'_k = \left(x'_k, y'_k, t_{ref}, p_k\right)\right\}_{k=1}^N$. Each event is aligned from t_k to t_{ref} along the motion curve that passes through it. Such alignment process can be described as:

$$(x'_k, y'_k) = (x_k, y_k) + (t_k - t_{ref})\mathbf{v}(x_k, y_k). \tag{2}$$

Then, aligned events $\varepsilon'_{t_{ref}}$ are aggregated on an image of warped events (i.e., appearance tensor):

$$T_a(x, y, \varepsilon'_{t_{ref}}) = \sum\nolimits_{k=1}^{N} \delta(x - x'_k)\delta(y - y'_k), \tag{3}$$

where each pixel (x, y) sums the number of warped events (x', y') that fall within it. $\delta(\cdot)$ denotes the Dirac delta function.

B. Fusion: Event Motion Guided Attention Module

Next, we introduce how to effectively integrate motion and appearance information and extract more effective features.

1) Overall Framework: The spatial and temporal feature interaction and fusion module is shown in Fig. 3(b), which consists of an appearance branch, a motion branch, and a set of event motion-guided attention modules (EMGA) that bridges the appearance and motion branch. The architecture of the appearance branch and motion branches are similar but slightly different. The motion branch utilizes a lighter design than the appearance since the motion tensor does not contain as many high-level semantics and subtle boundaries as the appearance tensor. More specifically, as can be seen in Fig. 3(b), the motion tensor T_m is first sent to the motion branch to generate the motion features $F_m^i (i = 0, 1, 2, 3)$. Then, the appearance tensor T_a is fed to the residual block ResBlock-0 of the appearance branch to extract the appearance feature F_a^0, which is then sent to the EMGA module together with the motion features F_m^0. In the EMGA module, the motion feature serves as guidance to emphasize important locations and channels within the appearance feature. Next, the motion-attended appearance feature F_{am}^0 is used as the input to the ResBlock-1 of the appearance branch to generate appearance features F_a^1. This process is repeated four times to extract visual features from low to high levels, obtaining four motion-attended appearance features $F_{am}^i (i = 0, 1, 2, 3)$ at different scales.

2) Event Motion Guided Attention Module: As shown in Fig. 4, the EMGA module is a residual model with lightweight spatial and channel attention, which takes a motion feature F_m^i and an appearance feature F_a^i as input. This module first integrates motion features with appearance features and then transforms them into spatial attention. In order to enhance key locations and elements within the appearance features, this module applies channel attention to enhance some latent attributes of the motion-attended appearance features and finally incorporates the input feature as a complement. As in the residual unit, there are two pathways: one is the identity path, and the other path is formed by spatial attention and channel attention. The EMGA can be further re-written as follows:

$$\begin{aligned}
\psi_s &= \text{Sigmoid}(h(\text{Concat}(F_m^i, F_a^i))),\\
F_{as}^i &= \psi_s \odot F_a^i,\\
\psi_c &= \text{Softmax}(h'(\text{GAP}(F_{as}^i))),\\
F_{am}^i &= F_a^i + F_{as}^i \odot \psi_c,
\end{aligned} \tag{4}$$

where $\odot$ represents element-wise multiplication. Both $h(\cdot)$ and $h'(\cdot)$ are implemented as 1×1 convolution whose output channels are 1 and C, respectively. $\text{GAP}(\cdot)$ denotes global average pooling in the spatial dimensions. ψ_s and ψ_c represent the spatial and channel attention coefficients, respectively.

With the help of this attention-like feature fusion approach, the EMGA module works well since it allows the network to focus on regions of interest and important channels for the following detection, segmentation, and pose estimation tasks. Moreover, the residual connection preserves the details and semantic information in the original features, complementing the features that may be incorrectly suppressed by spatial attention ψ_s or channel attention ψ_c. Thus, the residual formulation is used to enhance prominent motion parts without discarding rich spatial features. This strategy significantly enhances the robustness of our method.

C. Necks and Heads for Different Tasks

1) Object Detection: The object detection neck and head are composed of Feature Pyramid Net (FPN) and detection head, as shown in Fig. 3(c). Following the feature extraction module, a feature pyramid net is employed to aggregate the multi-layer features $F_{am}^i (i = 0, 1, 2, 3)$. The shallow-layer

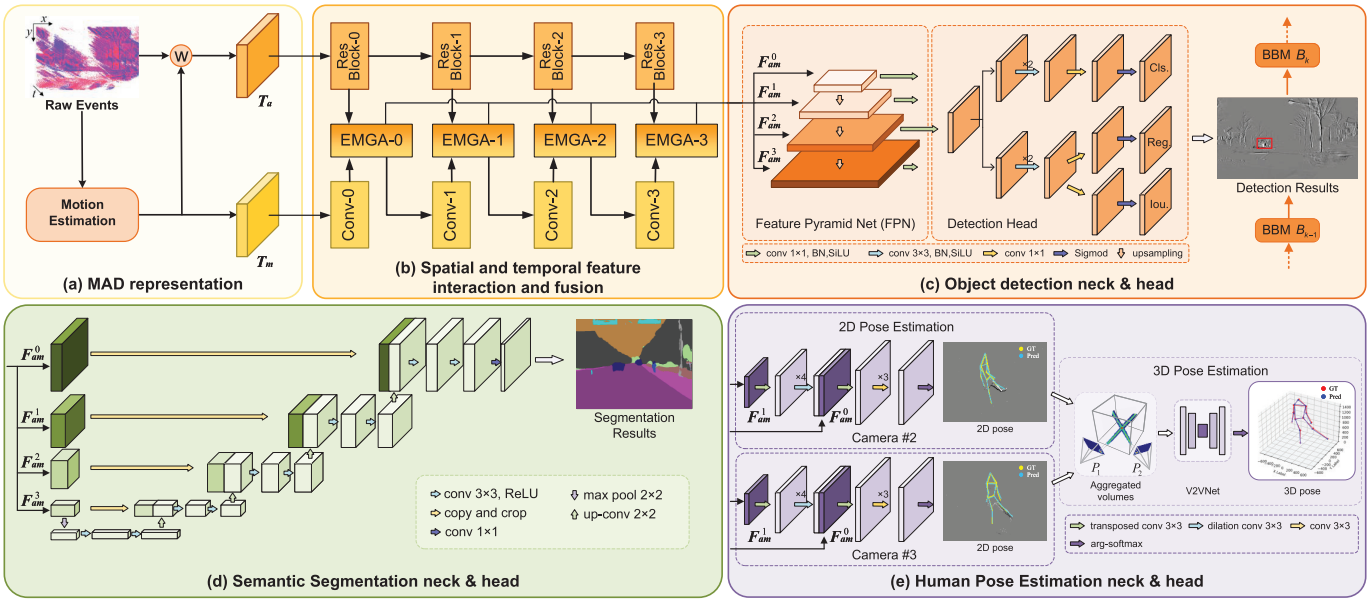


Fig. 3. The pipeline of the proposed MAD for different event-based applications. (a) The initial events data are first decoupled into motion tensor T_m and appearance tensor T_a and fed into two distinct convolutional branches. (b) Then, three high-level tasks share the same backbone with the EMGA module, which integrates features from both branches and achieves further feature interaction and fusion. Finally, these extracted features are fed into several downstream task-specific necks and heads (i.e., (c) FPN-based decoder for object detection, (d) Unet-based decoder for semantic segmentation, and (e) dual stream-based decoder for human pose estimation) to facilitate diverse event camera applications.

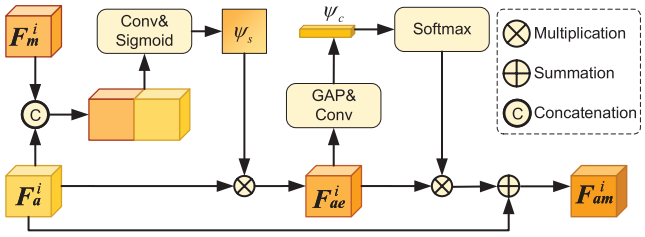


Fig. 4. Illustration of the Event Motion Guided Attention module (EMGA) module.

feature with rich spatial fine-grained information and the deep-layer feature with rich semantic information are concatenated to generate robust global feature maps. Subsequently, feature maps are fed to the detection head for final output. In addition, to detect targets in static scenes, we propose a simple, low-memory, and training-free mechanism called bounding box memory module (BBM) for event-based object detectors. It utilizes the fact that objects detected in one frame can not disappear without event generation. By comparing bounding the IoU of bounding boxes and the event density across frames, we can identify stationary objects missed by the detector due to low event activity and include them in the next frame's detection results.

2) Semantic Segmentation: For the semantic segmentation task, we replace the encoder in U-Net [44] with our specially designed backbone network for feature extraction, and maintain the same skip connections and decoder of Unet as shown in Fig. 3(d). Specifically, the input of this network consists of a set of feature maps $F^i_{am}(i = 0, 1, 2, 3)$ at different scales. These feature maps gradually enrich in semantic content but decrease in spatial resolution. In each upsampling step, the size of the feature map increases, and it fuses with the corresponding scale feature map through skip connections, which helps retain more detailed information.

3) Human Pose Estimation: Similar to previous works [22], [45], we decompose the three-dimensional pose estimation problem into two sub-problems: 2D pose estimation and 2D-to-3D pose reconstruction. As shown in Fig. 3(e), a single CNN model [22], [33] is first trained on the two camera views. A pose estimation network [45] is used to perform the 2D-to-3D reconstruction. This network takes a series of intermediate features from the 2D pose estimation network as input. Then, these features are unprojected into volumes with subsequent aggregation to a fixed-size volume. The volume is then passed to a 3D convolutional neural network (i.e., V2Vnet [46]), which outputs the interpretable 3D heatmaps. Finally, the 3D positions of joints can be inferred from these 3D joint heatmaps after soft argmax operation.

IV. Experiments

In this section, we first introduce the experimental setup, including the dataset and the implementation details. Then, we compare the proposed method with state-of-the-art methods. Afterward, we conduct ablations to validate our design choices and parameter settings. Finally, we conduct experiments on semantic segmentation and human pose estimation tasks to demonstrate the generalizability of the decoupling representation on different vision applications.

A. Experimental Settings

1) Datasets: We conduct experiments on three tasks across five datasets, including two object detection datasets (GEN1 Automotive Dataset [20] and Prophesee 1Mpx Detection

Dataset [26]), two semantic segmentation datasets (DSEC-Semantic Dataset [21] and DDD17 Datasets [3]), and a human pose estimation dataset (DHP19 Dataset [22]).

Object detection datasets. The GEN1 Automotive Dataset [20] comprises 39 hours of event data captured by an ATIS camera (resolution: 304 × 240). It includes manually annotated bounding boxes for 226,719 cars and 27,658 pedestrians, with annotation frequencies of 1 Hz, 2 Hz, or 4 Hz. Following previous works [47], [48], we remove the bounding boxes with short sides (i.e., less than 10 pixels) and diagonals (i.e., less than 30 pixels).

The 1Mpx Detection Dataset [26] provides event streams with a resolution of 1280 × 720, annotated at a frequency of 60 frames per second (FPS). This dataset includes 11.19 hours of training data, 2.21 hours of validation data and 2.25 hours of test data. This dataset contains a total of 7 categories. Following previous works [26], [47], [48], we utilize three categories (i.e., cars, pedestrians, and two-wheeled vehicles) for performance comparison. We ignore bounding boxes with a diagonal size less than 60 pixels or width or height less than 20 pixels during training and evaluation. For both object detection datasets, mean average precision (mAP) [49] and runtime are used as evaluation metrics. Unless otherwise specified, runtime measurements include both event representation generation and subsequent processing.

Semantic segmentation datasets. The DDD17 dataset [3] is a widely-used autonomous driving semantic segmentation dataset, containing 12 hours of driving data recorded by DAVIS. It provides pixel-wise aligned and temporally synchronized event and grayscale images, which belong to 6 labels: flat (road and pavement), background (construction and sky), object, vegetation, human, and vehicle.

The DSEC-Semantic dataset [21], collected from various urban and rural environments in Switzerland, consists of 8082 labeled frames for training and 2809 labeled frames for testing. All the scenes in this dataset are divided into 11 categories including: background, building, fence, person, pole, road, sidewalk, vegetation, car, wall, and traffic sign. Compared with the DDD17 dataset, the DSEC-Semantic dataset has higher resolution (resolution 440 × 640 vs. 352 × 200) and more fine-grained categories (i.e., 11 categories of labels vs. 6 categories of labels).

Human pose estimation datasets. The DHP19 dataset [22] is a real-world public dataset for human pose estimation. Its data is collected in a clean indoor environment using four synchronized cameras (346×280 pixels) placed at 90 degrees, −45 degrees, 45 degrees, and 90 degrees relative to the front of the subjects. It provides three-dimensional annotations and the camera parameters required to calculate two-dimensional projections. Recordings are composed of 17 subjects, each performing 33 actions. Following previous works [22], [33], we only used data from two front camera views.

2) Implementation Details: The training of our method is divided into two steps. First, we train the optical flow estimation network which is implemented as EV-Flownet [19]. Then, we freeze the parameters of EV-Flownet and train the rest of the network (e.g., the backbone, neck, and head) for different downstream tasks.

For the EV-Flownet, we randomly initialize all model layers and iterate for 100 epochs with the Adam [50] optimizer using a linear decay learning rate schedule. The initial learning rate of the GEN1 Automotive Dataset and DHP19 dataset is set to 1e-4. For the other three datasets, the initial learning rate is set to 5e-5. The batch size is set to 4 for all datasets.

For object detection experiments, we construct event representations within non-overlapping sliding time windows, with each time window being fixed at 50 *ms*. For the rest of the network, the appearance branch is initialized using an ImageNet dataset-pretrained ResNet [51] model, while the motion branch is randomly initialized. The initial learning rate for the GEN1 Automotive Dataset is set to 1e-4 with a batch size of 32. For the 1Mpx Detection Dataset, we set the initial learning rate to 5e-5 and batch size to 16. The learning rate linearly decays until it hits 1/10 of the initial value. Moreover, we utilize the same data augmentation methods as RVT [47] to prevent overfitting. We select the best training model on the validation dataset and evaluate it on the testing dataset. The mAP and Runtime in Table II are directly referred from publications [35], [47], [48], [52]. The runtime is measured in milliseconds for a batch size of 1 and includes the time of event representation. We used an NVIDIA TITAN Xp GPU for inference to compare against indicated timings in prior work [35], [47], [48], [52].

For semantic segmentation experiments, the rest of the network is trained from scratch on two datasets (i.e., DDD17 dataset and DSEC-Semantic dataset), respectively. The initial weights of the network are randomly generated. The learning rate is set to 0.002 and the batch size is set to 8 for both datasets. Our model is trained for 200 epochs, using the Adam [50] optimizer with Nesterov momentum being set to 0.9. For the DSEC-Semantic dataset, we select the best training model on the validation dataset and evaluate it on the testing dataset. Since the DDD17 dataset is only divided into training and test sets, we select the weights with the lowest loss on the training set as the final model weights. Consistent with previous works [53], the segmentation performance is evaluated by two commonly used metrics including Accuracy (ACC) and mean Intersection Over Union (mIoU). The ACC and mIoU in Table III are directly referred from publications [53], [54]. The runtime is obtained by reproducing the public code. Runtime is measured in milliseconds for a batch size of 1 and includes the time of event representation. We used an NVIDIA 3090 GPU for inference.

For human pose estimation experiments, we follow the original split setting of the DHP19 dataset [22], where the network training is conducted on S1-S12 and its evaluation is performed on S13-S17. Also, we only used data from the two front camera views (i.e., camera #2 and camera #3), same as [22] and [33]. We employ the Adam [50] optimizer with an initial learning rate of 1e-4, which decreases linearly to 1e-5 after 100 epochs. Mean per joint pixel error (MPJPE) is selected as the evaluation metric, i.e.,

$$\text{MPJPE} = \frac{1}{J}\sum_{i}^{J} \|pred_i - gt_i\|_2, \tag{5}$$

TABLE II
DETECTION PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART METHODS

Methods	Publication	Event Representation	Detection Neck & Head	Temporal	Gen1 Automotive		1Mpx Detection	
					mAP(%)	Runtime(ms)	mAP(%)	Runtime(ms)
NVS-S [10]	ICCV 2021	Graph	YOLOv1	No	8.6	-	-	-
ASYNet [55]	ECCV 2020	Point Cloud	YOLOv1	No	14.5	-	-	-
AEGNN [17]	CVPR 2022	Graph	YOLOv1	No	16.3	-	-	-
YOLOv3 Events [11]	ICRA 2019	Event Count	YOLOv3	No	31.2	22.3	34.6	49.4
E2Vid [12]	TPAMI 2019	Reconstructions	SSD	No	36.8	237	39.3	513
NGA-events [13]	ECCV 2020	Voxel Grid	YOLOv3	No	35.9	26.1	37.8	55.4
GET* [35]	ICCV 2023	Token	YOLOX	No	38.7	15.9	40.6	17.1
ASTMNet* [52]	TIP 2022	Asynchronous attention embedding	SSD	No	39.6	28.6	42.6	59.8
MAD-Det (Ours)*	-	MAD	YOLOX	No	**42.2**	**14.2**	**44.4**	**16.1**
RED [26]	NIPS 2022	Voxel Grid	SSD	Yes	40.0	16.7	43.0	39.3
ASTMNet [52]	TIP 2022	Asynchronous attention embedding	SSD	Yes	46.7	35.6	48.3	72.3
RVT [47]	CVPR 2023	Voxel Grid	YOLOX	Yes	47.2	**10.2**	47.4	**11.9**
GET [35]	ICCV 2023	Group Token	YOLOX	Yes	47.9	16.8	48.4	18.2
SAST [48]	CVPR 2024	Voxel Grid	YOLOX	Yes	48.2	18.2	48.7	19.7
MAD-Det (Ours)	-	MAD	YOLOX	Yes	**49.2**	15.2	**49.5**	17.2

A star * represents the results without memory enhancement. The **bold** and the underline represent the best and second-best performance, respectively.

TABLE III
SEMANTIC SEGMENTATION PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART METHODS

Methods	Publication	Event Representation	DDD17			DSEC-Semantic		
			ACC(%) ↑	mIoU(%)↑	Runtime(ms)	ACC(%)↑	mIoU(%)↑	Runtime(ms)
Ev-SegNet (C=2) [59]	CVPRW 2019	2-channel Image	88.85	53.07	36.5	87.56	50.11	40.2
Ev-SegNet (C=6) [59]	CVPRW 2019	6-channel Image	89.75	54.81	37.2	88.61	51.76	41.3
E2ViD [12]	TPAMI 2019	Reconstructions	85.84	48.47	103	80.06	44.08	198
Vid2E [60]	CVPR 2020	EST	90.19	56.01	197	80.32	44.86	371
EV-SegFormer (C=2) [53]	TIP 2023	2-channel Image	92.98	53.26	26.7	-	-	-
EV-SegFormer (C=6) [53]	TIP 2023	6-channel Image	**94.72**	54.41	27.5	-	-	-
HMNet-B [54]	CVPR 2023	ATS	-	-	-	88.70	53.80	**17.9**
HMNet-L [54]	CVPR 2023	ATS	-	-	-	89.80	55.00	20.7
MAD-Seg (Ours)	-	MAD	94.32	**56.91**	**16.6**	**90.22**	**55.92**	20.1

The **bold** and the underline represent the best and second-best performance, respectively.

TABLE IV
HUMAN POSE ESTIMATION PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART METHODS ON DHP19 DATASET

Method	Publication	MPJPE$_{2D}$ ↓	MPJPE$_{3D}$ ↓	Runtime(ms)
Ras-PointNet [56]	3DV 22	7.29	82.46	**12.29**
DHP19 [22]	CVPRW 19	7.67	87.90	27.55
DGCNN [63]	TOG 19	6.83	77.32	127.96
MobileHP-S [64]	CVPRW 21	5.65	64.14	48.09
LeVit-128S [65]	ICCV 21	7.68	87.79	14.40
PointTrans [66]	ICCV 21	6.46	73.37	497.27
VMVPointT [67]	RA-L 22	9.13	103.23	-
MoveEnet [58]	CVPRW 23	6.28	-	-
TORE [33]	TPAMI 23	5.77	58.40	35.32
VMST-Net [57]	TCSVT 24	6.45	73.04	-
MAD-Pose (Ours)	-	**5.02**	**56.32**	23.19

The **bold** and the underline represent the best and second-best performance, respectively.

where $pred_i$ and gt_i represent the prediction and ground truth of the i-th joint, respectively. J refers to the total number of joints. Pixels and millimeters are 2D and 3D error measurement units, respectively. The MPJPE$_{2D}$ ↓, MPJPE$_{3D}$ ↓ and runtime in Table IV are directly referred from publications [33], [56], [57], [58]. Runtime is measured in milliseconds for a batch size of 1 and includes the time of event representation. We used an NVIDIA 3090 GPU for inference.

B. Comparison to State-of-the-Art

1) Results on Object Detection: To demonstrate the superiority of our method, we compare our MAD-Det network to state-of-the-art (SOTA) event object detection methods, including NVS-S [10], ASYNet [55], AEGNN [17], YOLOV3 Event [11], E2Vid [12], NGA-events [13], GET [35], ASTMNet [52], RED [26], RVT [47], SAST [48].

Quantitative Results: Table II shows that our MAD-Det performs best on both Gen1 and 1Mpx datasets. More specifically, on the Gen 1 dataset, our method outperforms the second-best method (i.e., SAST) by 1.0% mAP (49.2% vs. 48.2%). On the 1MPX dataset, our method outperforms the second-best method (i.e., SAST) by 0.8% mAP (49.5% vs. 48.7%). Besides, when we did not utilize memory modules for enhancement as previous methods [35], [52], our method still achieves state-of-the-art performance on both datasets, even better than some networks with memory mechanism (e.g., RED (42.2% vs. 40.0%)). The superior performance of our method is mainly attributed to the decoupling representation,

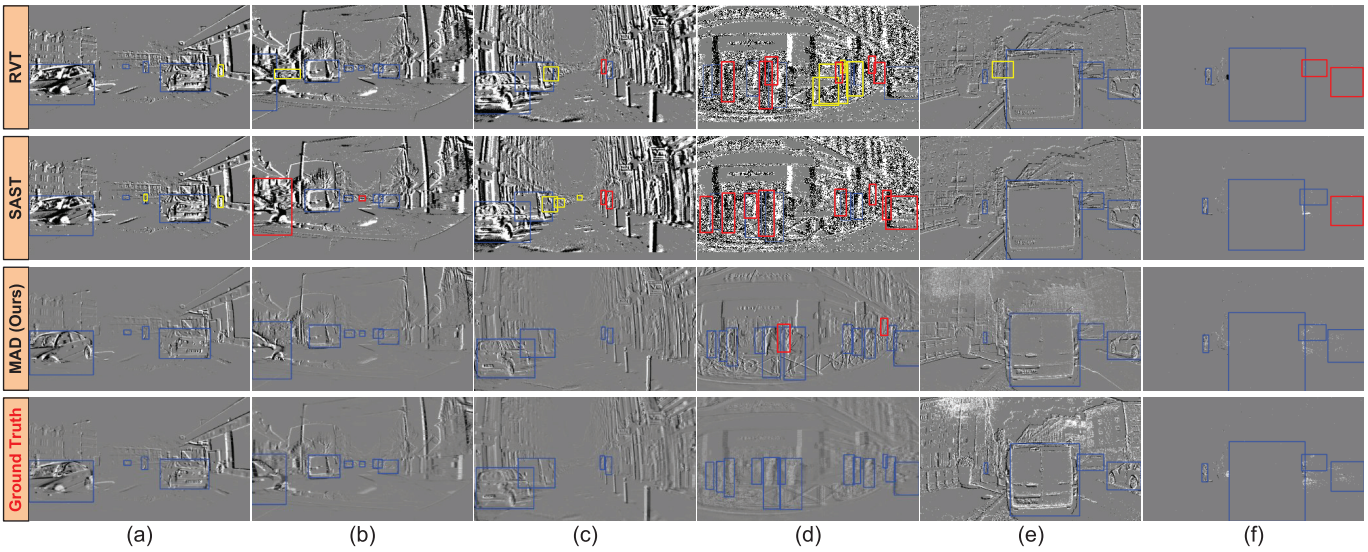


Fig. 5. Visualization results on the 1Mpx Detection Dataset [26] (resolution 1280 × 720). The correct detections, false alarms, and missed detections are highlighted with blue, yellow, and red boxes, respectively. The first and second rows represent the detection results of RVT [47] and SAST [48], respectively. The third row represents the detection results of our method. Our method obtains better performance than two competitors. (e) and (f) represent two moments within a consecutive static scene, where (e) is the moment right after becoming static, while (f) is the moment after being static for 20 seconds.

which allows the network to learn more discriminative features, and the EMGA module achieves better interaction and fusion of motion and appearance features.

Qualitative Results: Visualization results on the 1 Mpx Detection Dataset are shown in Fig. 5. Note that our method achieves the best performance among several state-of-the-art methods, including RVT [47] and SAST [48]. For example, RVT and SAST cannot accurately detect fast-moving targets in complex scenes, while our method can effectively detect them due to the reconstruction of clear contours of the scene (see in Figs. 5 (c) and (d)). In addition, results in Fig. 5(e) and (f) show that LSTM-based models can retain information over some period when no events are available. However, the memory of the network will decay over time, leading to a deterioration in detection performance. In contrast, our method exhibits significantly enhanced long-term memory retention capabilities, consequently delivering more robust and stable detection performance.

2) Results on Semantic Segmentation: We compare our MAD-seg with state-of-the-art event-based semantic segmentation methods, including Ev-SegNet [59], E2ViD [12], Vid2E [60], EV-SegFormer [53] and HMNet [54].

Quantitative Results: As shown in Table III, our MAD-Seg achieves state-of-the-art performance on the DSEC-Semantic dataset. Compared to the second top-performing method, our method improves ACC by 0.42% (90.22% vs. 89.80%) and mIoU by 0.92% (55.92% vs. 55.00%). On the DDD17 dataset, the ACC is slightly lower than that of EV-SegFormer (C=6), but the mIoU value of our method is significantly higher by 2.5% (56.91% vs. 54.41%). This result indicates that our method has higher accuracy and robustness in pixel-level classification. Different from previous methods [53], [54], [61], which specifically design a complex network structure for event semantic segmentation, we simply modify the Unet [44] and adapt it to our representation method and also achieve comparable performance, which demonstrates the potential of our proposed MAD representation in semantic segmentation task.

Qualitative Results: Fig. 6 visualizes the segmentation results on the DDD17 datasets. Results show that our method can predict smaller targets, such as traffic signs and pedestrians in the distance while exhibiting fewer artifacts, wrong results, and missing objects. That is because, decoupling representation reconstructs clear contours and boundaries of the target and provides additional motion cues (i.e., the same object usually exhibits consistent movement). This makes it easier for the network to capture slight differences between different objects.

3) Results on Human Pose Estimation: Human pose estimation is a critical application for event cameras. Motion blur caused by rapid movements like waving arms or jumping affects the accuracy of human pose estimation in traditional camera images. Event cameras provide a solution for addressing rapid human movements. Consequently, more studies are adopting event cameras for human pose estimation tasks. In this paper, we compare our MAD-Pose with state-of-the-art event-based human pose estimation methods, including RasPointNet [56], [62], DHP19 [22], DGCNN [63], MobileHP-S [64], LeVit-128S [65], PointTrans [66], VMVPointT [67], MoveEnet [58], TORE [33] and VMST-Net [57].

Quantitative Results: Table. IV shows the results of the human pose estimation experiments. Compared with the second top-performing method TORE [33], the $MPJPE_{2D}$ and $MPJPE_{3D}$ of our approach decreases by 0.75 pixels (5.02 vs. 5.77) and 2.08 millimeters (56.32 vs. 58.40), respectively. This result strongly demonstrates the effectiveness of our method.

Qualitative Results: The visualization results presented in Fig. 7 illustrate the results of our pose estimation approach. Our model demonstrates superior capability in delivering

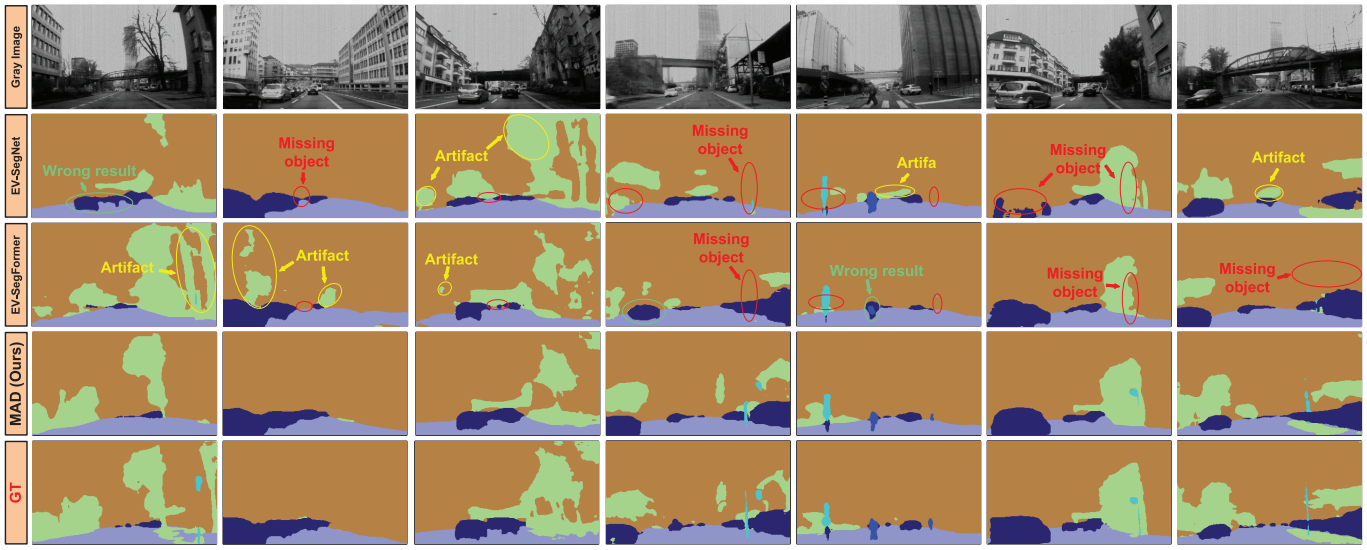


Fig. 6. Visual comparison with other methods on the DDD17 dataset. Gray images are shown for visualization only. Missing objects, artifacts, and wrong results are highlighted by red, yellow, and green dotted circles, respectively. Compared to EV-SegNet and EV-SegFormer, our method can predict smaller targets, such as traffic signs and pedestrians in the distance, and exhibit fewer artifacts, wrong results, and missing objects.

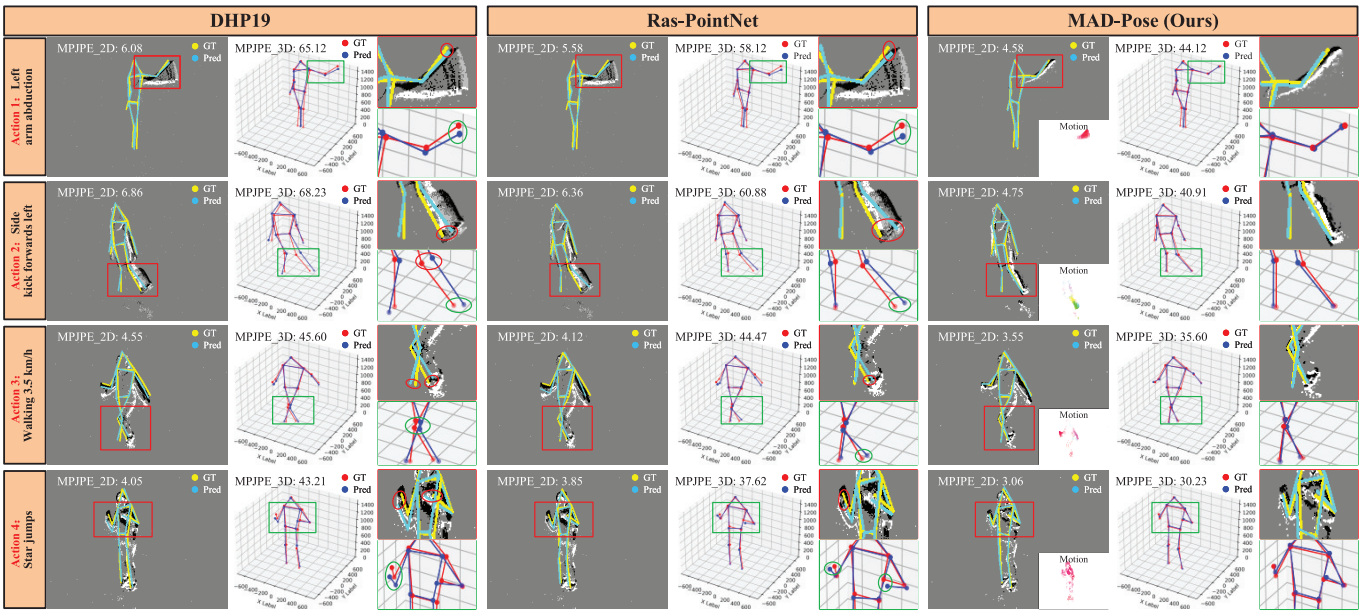


Fig. 7. Visual comparison with other methods on the DHP19 dataset. The first to third columns are the 2D pose estimation results (yellow for ground truth, blue for prediction), 3D pose estimation results (red for ground truth, blue for prediction), and the detailed results of the major moving parts of the body (marked by boxes), respectively. The fourth to sixth columns are the results of Ras-PointNet, and the seventh to last columns are the results of our method. Our method achieves great results on four different movements, including "Left arm abduction", "Side kick forwards left", "Star jumps" and "Walking 3.5 km/h" from top to bottom.

high-precision estimations for the majority of joints, particularly those exhibiting swift motion (highlighted with boxes). For instance, in the case of "Left arm abduction" (shown in the first row), other methods suffer from historical artifacts that lead to significant pose estimation errors, whereas our method accurately reconstructs the arm's position, achieving more precise pose estimation. By effectively mitigating historical artifacts induced by rapid movements, our approach restores clear articular structures. Thus, the proposed MAD representation proves particularly suitable for dynamic scenarios requiring finer motion details.

C. Ablation Study

1) Contribution of Our MAD Components: According to the previous analysis, our method achieves state-of-the-art performance on three different downstream tasks. As shown in Table V, we further explore the impact of each component on the ultimate performance. It can be observed that using only the appearance tensor as input significantly improves performance as compared with the event count image (e.g., 80.38% ACC vs. 92.96% ACC on the DDD17 dataset). This is because the clear contours and shapes of the appearance

TABLE V
THE PERFORMANCE OF OUR METHOD COMPONENTS ON FIVE DATASETS

Motion	Appearance	EMGA	Gen1	1Mpx	DDD17		DSEC-Semantic		DHP19	
			mAP(%)↑	mAP(%)↑	ACC(%)↑	mIoU(%)↑	ACC(%)↑	mIoU(%)↑	$MPJPE_{2D}$ ↓	$MPJPE_{3D}$ ↓
			39.4	40.4	80.38	42.11	81.36	40.39	9.69	102.37
✓			15.3	13.2	43.66	15.32	46.93	12.23	39.29	432.32
	✓		47.9	48.2	92.96	55.23	88.60	54.32	5.92	62.36
✓	✓		48.1	48.4	93.67	55.72	89.32	54.83	5.53	60.49
✓	✓	✓	49.2	49.5	94.32	56.91	90.22	55.92	5.02	56.32

The baseline model takes the event count image as input for the appearance branch. For the motion branch, it uses an array filled with zeros. It uses concatenation as the feature fusion method.

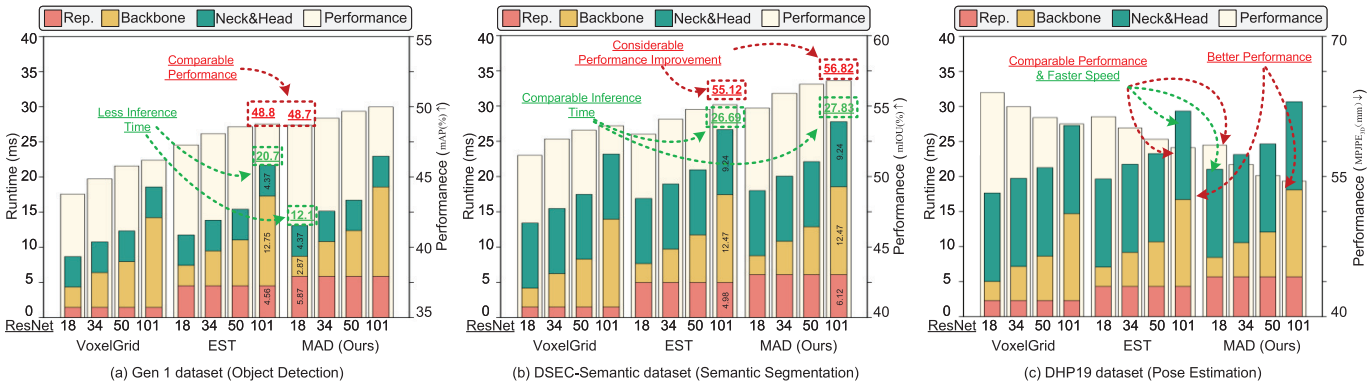


Fig. 8. Comparison of different event representation methods (i.e., Voxle Grid [30], EST [32] and our proposed MAD) on different backbones (i.e., Resnet18, Resnet34, Resnet50, and Resnet101). For all event representation methods, we adopt the same network architecture as shown in Fig 3. For EST and Voxel Grid, we set the motion tensor as an array filled with zeros. We report the runtime of three stages of the networks: representation computation, feature extraction (i.e., backbone), and neck & head processing. The runtime for representation computation contains the entire duration from the raw event data to the final "intermediate representation". MAD introduces minimal computational overhead during representation yet achieves comparable performance using shallow backbones relative to other methods combined with deeper backbones, thus significantly reducing overall running time.

information help the network learn more distinctive features. When using only the motion tensor as input, the performance is poor, which is even far below the baseline (e.g., 15.3% mAP vs. 39.4% mAP on the Gen 1 dataset). This is because motion information can only provide rough scene motion cues and cannot represent detailed features such as textures and contours with rich semantics. However, the motion tensor is not useless. It can serve as an auxiliary clue to enhance the final performance. By simply concatenating the motion tensor with the appearance tensor, the network performance can be slightly improved (e.g., 0.72% ACC increase on the DSEC-Semantic dataset). Furthermore, by integrating specially designed EMGA modules, the performance of our method is further improved (e.g., 48.1% mAP increased to 49.2% mAP on the Gen 1 dataset). This indicates that the proposed EMGA module can effectively integrate appearance and motion features.

2) The Effectiveness of Decoupling Representation: We conduct an ablation experiment to study the effectiveness of motion and appearance decoupling representation. As shown in Fig. 8, we report the performance of different representation methods with various depth feature extraction networks and the inference time for each component of the network. Note that, with the same feature extraction network, MAD consistently outperforms other representation methods. For example, as shown in Fig. 8(b), when combined with ResNet-101 backbone, MAD achieves significantly better performance than EST (56.82% vs. 55.12%). This indicates that the MAD representation improves the performance upper bound, which is crucial in scenarios requiring high precision.

Much to our surprise, our MAD with a shallow backbone achieves comparable performance with other representation methods with a deeper backbone. For instance, Fig. 8(a) shows that MAD with ResNet-18 achieves comparable object detection performance to EST with ResNet-101 (48.7% vs. 48.8%), while reducing total runtime by 9.88 ms (12.1*ms* vs. 20.7*ms*). Although compared with EST, MAD spends an additional 1.31ms (5.87*ms* vs. 4.56*ms*) in the representation phase, the runtime savings from using a shallow backbone are substantially greater (2.87*ms* vs. 12.75*ms*). These results demonstrate that MAD reduces overall runtime despite its representation overhead, while maintaining competitive performance. The improvement is because that our MAD decouples the highly entangled spatial-temporal event streams into semantically distinct appearance tensors and motion tensors, thereby enabling the network to learn discriminative spatial and temporal features more effectively.

3) The Influence of Motion Estimation Accuracy: We conduct an ablation experiment to study the influence of motion estimation accuracy on the final performance. This experiment is exclusively conducted on the DSEC dataset, as it provides both semantic segmentation ground truth and optical flow

TABLE VI
INFLUENCE OF MOTION ESTIMATION ACCURACY ON THE DSEC-SEMANTIC DATASET

Method	Sup.	EPE(px)↓	mIoU(%)↑	Rep. Time(ms)
GT	-	0	56.23	-
IDNet(4×)	Yes	0.72	56.13	87.6
E-RAFT	Yes	0.81	56.11	47.3
Ev-Flownet	No	3.86	55.92	6.12
Firenet	No	4.53	55.75	5.69

Rep. Time includes the time of optical flow estimation and event alignment. Sup. represents supervised methods.

TABLE VII
COMPARISON WITH DIFFERENT FUSION METHODS

Method	Gen1	1Mpx	DDD17	DSEC	DHP19
Concatenation	48.1	48.4	55.01	54.77	65.72
Addition	47.5	47.3	55.21	54.68	68.72
Multiplication	48.4	48.6	55.03	54.53	65.42
Cross attention	48.9	49.3	55.93	55.11	59.58
EAMG	33.2	32.2	34.92	33.32	112.31
EMGA	49.2	49.5	56.91	55.92	56.32

Values represent mAP(%) for Gen1 and 1Mpx dataset, mIoU(%) for DDD17 and DSEC dataset, and MPJPE$_{3D}$(mm) for DHP19 dataset.

annotations, whereas other datasets only contain task-specific labels. Specifically, we replace the "motion estimation" in Fig. 3(a) with different optical flow estimation networks, including two unsupervised methods, Ev-Flownet [19] and Firenet [68], and two fully supervised methods, IDNet [69] and E-RAFT [70]. As shown in Table. VI, as the optical flow estimation error decreases, the performance tends to keep increasing. This indicates that the more accurate the optical flow is, the more precise the spatial-temporal information contained in the appearance tensor and motion tensor will be, which results in better segmentation performance. However, increasing the accuracy of optical flow estimation also comes with increased computational complexity and memory requirements. And higher accuracy requires the ground truth of optical flow as supervision, which is hard to get. Therefore, it is reasonable to use Ev-Flownet as the motion estimation algorithm to achieve a trade-off. Notably, When using the simplest and lightest optical flow estimation network, FireNet, which has a relatively high error (4.53 EPE) in complex scenes, the final semantic segmentation performance (55.75 mAP) of MAD still outperforms the second-best performance method (i.e., HMNet-L with 55.00 mAP in Table III). This shows that even incomplete decoupling of event data using a fast optical flow estimation network can effectively improve the performance of the algorithm.

4) The Effectiveness of EMGA Module: To explore the effectiveness of our proposed EMGA, we compared it with other fusion methods, including concatenation, element-wise multiplication, addition, cross attention [71] and event appearance guided attention (EAGA), respectively. Specifically, the 'Concatenation' fusion first connects the C-channel appearance features and C'-channel motion features along the channel dimension and then applies a 1×1 convolution with C output channels. The 'Addition' fusion first adjusts the motion features to C channels through a 1×1 convolution, and then each element of the motion features is added to the corresponding element in the appearance features. The 'Multiplication' fusion works in a similar way to the 'Addition' fusion, only that addition has been replaced with multiplication. The 'Cross attention' weights the features based on the correlation between the appearance tensor and the motion tensor. The 'EAGA' represents for Event Appearance Guided Attention module, which shares the same structure as the 'EMGA' module but swaps the positions of motion tensor and appearance tensor. Each fusion method replaces 'EMGA' in Fig. 3. As shown in Table VII, the EMGA module obtains better performance than other fusion methods, which indicates that the proposed EMGA modules effectively integrate the appearance and motion branches.

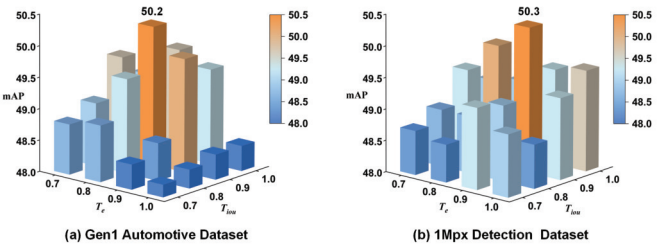


Fig. 9. The results of different threshold settings on the validation set.

5) Optimal Parameters of BBM: The BBM module includes two thresholds. The IoU threshold T_{iou} determines the overlap between two detection boxes. If the overlap is below this threshold, it is considered that the target will disappear in the next frame. The other threshold is the average event numbers threshold T_e, which assesses the average number of events within a detection box. If the average number of events is below this threshold, the disappearance of the target may be due to insufficient events. We test different combinations of T_{iou} and T_e on the validation set, and the results are shown in Fig. 9. On the Gen1 dataset, the best performance is achieved when T_{iou} is set to 0.8 and T_e is set to 0.9. On the 1Mpx dataset, the highest performance is obtained when T_{iou} is set to 0.9 and T_e is set to 0.9.

6) Failure Case Analysis and Discussion: Although our MAD achieves satisfying performance even in challenging scenarios, several limitations observed in failure cases require further investigation. As shown in Fig. 10, the semantic segmentation result illustrates the inherent difficulty in accurately representing small targets. While our method successfully reconstructs fine contour details, its performance degrades for diminutive objects due to their severely attenuated texture cues. This issue could potentially be mitigated by adopting a more powerful feature extraction backbone or a network architecture explicitly optimized for small-scale targets. Furthermore, the results on object detection and human pose estimation reveal that it is challenging to robustly represent static or slowly moving objects. This limitation stems from the fundamental operating principle of event cameras: although

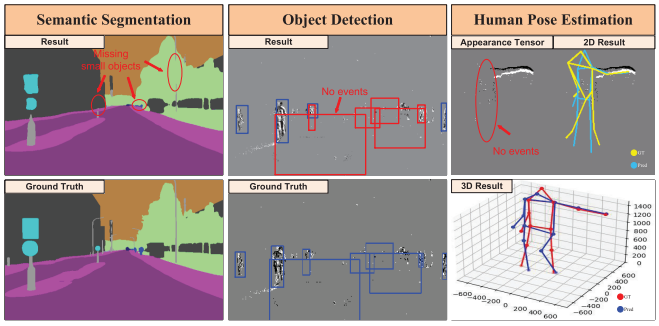


Fig. 10. Representative failure cases of our method in three downstream tasks. The first column demonstrates the limited performance of MAD in segmenting small targets. The second example illustrates the failure of MAD to detect long-term stationary objects. The correct detections and missed detections are highlighted with blue and red boxes, respectively. The third column reveals that when part of a human body moves slowly, MAD cannot accurately estimate the location of the corresponding joint.

they excel at sensing dynamic changes, they generate negligible events in static or quasi-static scenes. A promising solution to this problem lies in the fusion of event cameras with frame-based cameras [72] or vidar cameras [73], as they can provide complementary static scene information. Exploring such hybrid architectures presents a valuable direction for future research.

V. Conclusion

This paper introduces a novel event representation for decoupling motion and appearance. This bio-inspired design helps the network to extract discriminative temporal (i.e., motion) and spatial (i.e., appearance) information and thus reduces the learning burden of the network toward complex high-level interpretation tasks. Then, we design a dual-stream detection network and event motion-guided attention (EMGA) module to achieve temporal and spatial feature interaction and fusion. Finally, three specially designed decoder heads are proposed for three representative event-based tasks (i.e., object detection, semantic segmentation, and human pose estimation). The experimental results demonstrate that our method performs better than the state-of-the-art methods on all these three tasks. We hope this work will draw attention to the research on this new event data representation method.

References

[1] 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 Process. Mag.*, vol. 37, no. 4, pp. 34–49, Jul. 2020.

[2] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5419–5427.

[3] J. Binas, D. Neil, S.-C. Liu, and T. Delbruck, "DDD17: End-to-end DAVIS driving dataset," 2017, *arXiv:1711.01458*.

[4] S.-C. Liu, B. Rueckauer, E. Ceolini, A. Huber, and T. Delbruck, "Event-driven sensing for efficient perception: Vision and audition algorithms," *IEEE Signal Process. Mag.*, vol. 36, no. 6, pp. 29–37, Nov. 2019.

[5] T. Delbruckl, "Neuromorophic vision sensing and processing," in *Proc. 42nd Eur. Solid-State Circuits Conf.*, Sep. 2016, pp. 7–14.

[6] J. Kim et al., "Privacy-preserving visual localization with event cameras," 2022, *arXiv:2212.03177*.

[7] M. Litzenberger et al., "Estimation of vehicle speed based on asynchronous data from a silicon retina optical sensor," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2006, pp. 653–658.

[8] S. B. Shrestha and G. Orchard, "SLAYER: Spike layer error reassignment in time," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 1419–1428.

[9] 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.

[10] Y. Li et al., "Graph-based asynchronous event processing for rapid object recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 914–923.

[11] Z. Jiang et al., "Mixed frame-/event-driven fast pedestrian detection," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8332–8338.

[12] 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.

[13] Y. Hu, T. Delbrück, and S. Liu, "Learning to exploit multiple vision modalities by using grafted networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 85–101.

[14] F. Paredes-Vallés, K. Y. W. Scheper, and G. C. H. E. De Croon, "Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2051–2064, Aug. 2020.

[15] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks," *IEEE Signal Process. Mag.*, vol. 36, no. 6, pp. 51–63, Nov. 2019.

[16] L. Cordone, B. Miramond, and P. Thierion, "Object detection with spiking neural networks on automotive event data," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2022, pp. 1–8.

[17] S. Schaefer, D. Gehrig, and D. Scaramuzza, "AEGNN: Asynchronous event-based graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12371–12381.

[18] M. Cannici, M. Ciccone, A. Romanoni, and M. Matteucci, "Asynchronous convolutional networks for object detection in neuromorphic cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1656–1665.

[19] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "EV-FlowNet: Self-supervised optical flow estimation for event-based cameras," 2018, *arXiv:1802.06898*.

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

[21] Z. Sun, N. Messikommer, D. Gehrig, and D. Scaramuzza, "ESS: Learning event-based semantic segmentation from still images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 341–357.

[22] E. Calabrese et al., "DHP19: Dynamic vision sensor 3D human pose dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1695–1704.

[23] R. Christoph and F. A. Pinz, "Spatiotemporal residual networks for video action recognition," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2016, pp. 3468–3476.

[24] T. Zhou, J. Li, S. Wang, R. Tao, and J. Shen, "MATNet: Motion-attentive transition network for zero-shot video object segmentation," *IEEE Trans. Image Process.*, vol. 29, pp. 8326–8338, 2020.

[25] S. Yang, L. Zhang, J. Qi, H. Lu, S. Wang, and X. Zhang, "Learning motion-appearance co-attention for zero-shot video object segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 1544–1553.

[26] E. Pérot, P. D. Tournemire, D. O. Nitti, J. Masci, and A. Sironi, "Learning to detect objects with a 1 megapixel event camera," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 16639–16652.

[27] H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 993–1000.

[28] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-based visual flow," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 407–417, Feb. 2014.

[29] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman, "HATS: Histograms of averaged time surfaces for robust event-based object classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1731–1740.

[30] 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.

[31] Y. Nam, M. Mostafavi, K.-J. Yoon, and J. Choi, "Stereo depth from events cameras: Concentrate and focus on the future," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6104–6113.

[32] D. Gehrig, A. Loquercio, K. Derpanis, and D. Scaramuzza, "End-to-end learning of representations for asynchronous event-based data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5632–5642.

[33] R. W. Baldwin, R. Liu, M. Almatrafi, V. Asari, and K. Hirakawa, "Time-ordered recent event (TORE) volumes for event cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 2519–2532, Feb. 2023.

[34] M. Cannici, M. Ciccone, A. Romanoni, and M. Matteucci, "A differentiable recurrent surface for asynchronous event-based data," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 136–152.

[35] Y. Peng, Y. Zhang, Z. Xiong, X. Sun, and F. Wu, "GET: Group event transformer for event-based vision," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 6038–6048.

[36] Z. Wang et al., "EAS-SNN: End-to-end adaptive sampling and representation for event-based detection with recurrent spiking neural networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024, pp. 310–328.

[37] J. Zhang et al., "Spiking neural networks with adaptive membrane time constant for event-based tracking," *IEEE Trans. Image Process.*, vol. 34, pp. 1009–1021, 2025.

[38] S. Wu, H. Sheng, H. Feng, and B. Hu, "EGSST: Event-based graph spatiotemporal sensitive transformer for object detection," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2024, pp. 120526–120548.

[39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[40] Q. Qu, X. Chen, Y. Y. Chung, and Y. Shen, "EvRepSL: Event-stream representation via self-supervised learning for event-based vision," *IEEE Trans. Image Process.*, vol. 33, pp. 6579–6591, 2024.

[41] Z. Liu, B. Guan, Y. Shang, Q. Yu, and L. Kneip, "Line-based 6-DoF object pose estimation and tracking with an event camera," *IEEE Trans. Image Process.*, vol. 33, pp. 4765–4780, 2024.

[42] P. Zhang et al., "Event-assisted blurriness representation learning for blurry image unfolding," *IEEE Trans. Image Process.*, vol. 33, pp. 5824–5836, 2024.

[43] Z. Sun, X. Fu, L. Huang, A. Liu, and Z.-J. Zha, "Motion aware event representation-driven image deblurring," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024, pp. 418–435.

[44] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, 2015, pp. 234–241.

[45] B. Jiang, L. Hu, and S. Xia, "Probabilistic triangulation for uncalibrated multi-view 3D human pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 14804–14814.

[46] J. Y. Chang, G. Moon, and K. M. Lee, "V2V-PoseNet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5079–5088.

[47] M. Gehrig and D. Scaramuzza, "Recurrent vision transformers for object detection with event cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 13884–13893.

[48] Y. Peng, H. Li, Y. Zhang, X. Sun, and F. Wu, "Scene adaptive sparse transformer for event-based object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 16794–16804.

[49] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 740–755.

[50] D. P. Kingma and J. L. Ba, "Adam: Amethod for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–15.

[51] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[52] 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.

[53] Z. Jia et al., "Event-based semantic segmentation with posterior attention," *IEEE Trans. Image Process.*, vol. 32, pp. 1829–1842, 2023.

[54] R. Hamaguchi, Y. Furukawa, M. Onishi, and K. Sakurada, "Hierarchical neural memory network for low latency event processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 22867–22876.

[55] N. Messikommer, D. Gehrig, A. Loquercio, and D. Scaramuzza, "Event-based asynchronous sparse convolutional networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 415–431.

[56] J. Chen, H. Shi, Y. Ye, K. Yang, L. Sun, and K. Wang, "Efficient human pose estimation via 3D event point cloud," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2022, pp. 1–10.

[57] D. Liu, T. Wang, and C. Sun, "Voxel-based multi-scale transformer network for event stream processing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 4, pp. 2112–2124, Apr. 2024.

[58] G. Goyal, F. Di Pietro, N. Carissimi, A. Glover, and C. Bartolozzi, "MoveEnet: Online high-frequency human pose estimation with an event camera," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2023, pp. 4024–4033.

[59] I. Alonso and A. C. Murillo, "EV-SegNet: Semantic segmentation for event-based cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, p. 0.

[60] D. Gehrig, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza, "Video to events: Recycling video datasets for event cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3583–3592.

[61] N. Messikommer, D. Gehrig, M. Gehrig, and D. Scaramuzza, "Bridging the gap between events and frames through unsupervised domain adaptation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3515–3522, Apr. 2022.

[62] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[63] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Oct. 2019.

[64] S. Choi, S. Choi, and C. Kim, "MobileHumanPose: Toward real-time 3D human pose estimation in mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 2328–2338.

[65] B. Graham et al., "LeViT: A vision transformer in ConvNet's clothing for faster inference," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12259–12269.

[66] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16239–16248.

[67] B. Xie, Y. Deng, Z. Shao, H. Liu, and Y. Li, "VMV-GCN: Volumetric multi-view based graph CNN for event stream classification," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1976–1983, Apr. 2022.

[68] J. J. Yu, A. W. Harley, and K. G. Derpanis, "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness," in *Proc. Eur. Conf. Comput. Vis. Workshops (ECCVW)*, 2016, pp. 3–10.

[69] Y. Wu, F. Paredes-Vallés, and G. C. H. E. De Croon, "Lightweight event-based optical flow estimation via iterative deblurring," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 14708–14715.

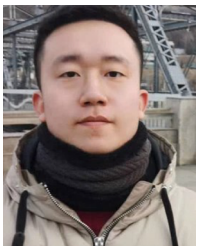
[70] M. Gehrig, M. Millhäusler, D. Gehrig, and D. Scaramuzza, "E-RAFT: Dense optical flow from event cameras," in *Proc. Int. Conf. 3D Vis. (3DV)*, Dec. 2021, pp. 197–206.

[71] H. Lin, X. Cheng, X. Wu, and D. Shen, "CAT: Cross attention in vision transformer," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2022, pp. 1–6.

[72] D. Li, Y. Tian, and J. Li, "SODFormer: Streaming object detection with transformer using events and frames," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 14020–14037, Nov. 2023.

[73] J. Li, X. Wang, L. Zhu, J. Li, T. Huang, and Y. Tian, "Retinomorphic object detection in asynchronous visual streams," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2022, vol. 36, no. 2, pp. 1332–1340.

Nuo Chen received the B.E. degree in mechanical design manufacture and automation and the M.S. degree in mechanical and electrical engineering from the Central South University (CSU), China, in 2019 and 2022, respectively. He is currently pursuing the Ph.D. degree in information and communication engineering with the National University of Defense Technology (NUDT), Changsha, China. His research interests include image processing, neuromorphic cameras, and computer vision.



Boyang Li received the B.E. degree in mechanical design manufacture and automation from Tianjin University, China, in 2017, the M.S. degree in biomedical engineering from the National Innovation Institute of Defense Technology, Academy of Military Sciences, Beijing, China, in 2020, and the Ph.D. degree in information and communication engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2023. He is currently an Assistant Professor with the College of Electronic Science and Technology, NUDT. His research interests include optical image processing, interpretation and application, particularly on infrared small target detection, weakly supervised semantic segmentation, and neural network compression and accelerating.

Yingqian Wang (Member, IEEE) received the B.E. degree in electrical engineering from Shandong University, Jinan, China, in 2016, and the master's and Ph.D. degrees in information and communication engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2018 and 2023, respectively. He is currently an Assistant Professor with the College of Electronic Science and Technology, NUDT. His research interests include light field image processing, image super-resolution, and infrared small target detection.

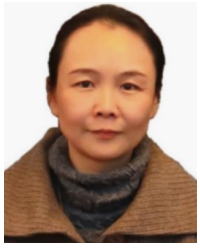
Xinyi Ying received the M.E. degree in information and communication engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2021. She is currently pursuing the Ph.D. degree with the College of Electronic Science and Technology, NUDT. Her research interests include detection and tracking of infrared small targets.

Longguang Wang received the B.E. degree in electrical engineering from Shandong University (SDU), Jinan, China, in 2015, and the Ph.D. degree in information and communication engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2022. His current research interests include low-level vision and 3D vision.

Chushu Zhang received the M.S. degree in electronic science from the National University of Defense Technology (NUDT), Changsha, China, in 2024. Her research interests include event cameras, low-level vision tasks, natural language processing, and lightweight inference for large language models (LLMs).

Yulan Guo (Senior Member, IEEE) received the B.E. and Ph.D. degrees from the National University of Defense Technology (NUDT) in 2008 and 2015, respectively. He has authored over 200 papers at highly referred journals and conferences. His research interests include 3D vision, low-level vision, and machine learning. He is a Senior Member of ACM. He served as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING. He also served the Area Chair for CVPR 2023/2021, ICCV 2021, ECCV 2023, and ACM Multimedia in 2021.

Miao Li received the M.E. and Ph.D. degrees from the National University of Defense Technology (NUDT) in 2012 and 2017, respectively. He is currently an Associate Professor with the College of Electronic Science and Technology, NUDT. His current research interests include infrared dim and small target detection.



Wei An received the Ph.D. degree from the National University of Defense Technology (NUDT), Changsha, China, in 1999. She was a Senior Visiting Scholar with the University of Southampton, Southampton, U.K., in 2016. She is currently a Professor with the College of Electronic Science and Technology, NUDT. She has authored or co-authored over 100 journal and conference publications. Her current research interests include signal processing and image processing.