Instance Segmentation as Tracking: A New Paradigm for Multi-Small-Object Tracking with Event Cameras

Nuo Chen[1], Shiman He[1], Boyang Li[1], Yingqian Wang[1], Chao Xiao[1], QianYin[1], Ruojing Li[1], Yihang Luo[1], Wei An[1], and Miao Li[1]*

National University of Defense Technology

Abstract. Multiple small object tracking (MSOT) is critical for applications such as anti-UAV systems and security surveillance, yet traditional frame-based cameras struggle to track fast-moving small objects in complex environments due to their low frame rates and limited dynamic range. Event cameras, with their ability to continuously record subtle brightness changes, can naturally overcome these limitations. However, most existing event-based tracking methods follow a "convert-then-detect-and-track" pipeline. This pipeline sacrifices the inherent high temporal resolution of event data, leading to fragmented trajectories of fast-moving objects. Moreover, it introduces significant background redundancy during framing, which reduces computational efficiency. To handle these issues, we introduce **"instance segmentation as tracking"**, a novel paradigm that formulates event-based MSOT as an instance segmentation task in the spatio-temporal dimension. Following this paradigm, we first design **Ev-ISNet**, which leverages 3D sparse convolutions to extract per-voxel features while simultaneously predicting object confidence and motion direction. Then we construct an event graph and progressively cluster trajectory instances by predicting edge-wise instance affinities. By leveraging intra-graph and inter-graph association modules, our method achieves highly efficient streaming inference. To address the lack of large-scale benchmarks for event-based MSOT, we build **EV-UAV-Track**, a comprehensive dataset featuring per-event instance-level annotations. Extensive experiments demonstrate that Ev-ISNet consistently outperforms state-of-the-art MSOT methods, achieving over 30% improvement in MOTA score compared to frame-based tracking methods.

1 Introduction

Multiple Small Object Tracking (MSOT) is of significant importance and has wide applications in areas such as anti-UAV systems, intelligent security, and autonomous monitoring [8, 33, 50]. Traditional MSOT systems primarily rely on infrared or RGB cameras [42]. However, constrained by their limited frame

* Corresponding author. Email: lisi@university.edu.cn

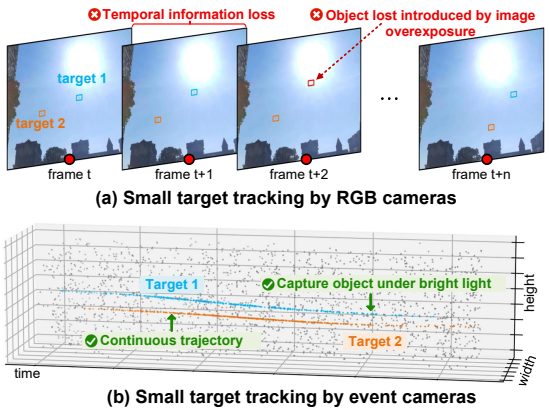

Fig. 1: Event camera vs. RGB camera. Conventional frame-based cameras, limited by their narrow dynamic range (≈60 dB) and low frame rates (typically 30-60 Hz), struggle with target tracking under complex illumination and high-speed motion. In contrast, event cameras leverage a high dynamic range (120 dB) and high temporal resolution (up to 10^6 Hz) to enable robust tracking in such challenging conditions.

rates and dynamic range, these conventional imaging modalities often fail to maintain stable and robust tracking performance under challenging conditions, including high-speed target motion, dynamic illumination changes, and cluttered backgrounds [19, 22, 23].

Event cameras have emerged as a novel bio-inspired sensing technology [4, 7, 9, 21]. Unlike traditional frame-based cameras that capture full images at fixed intervals, event cameras asynchronously output brightness change events, offering distinct advantages including microsecond-level temporal resolution, high dynamic range, and data sparsity [18, 28, 29]. These properties make them particularly suitable for complex scenarios involving high-speed motion and extreme lighting conditions, as shown in Fig. 1.

However, most existing event-based tracking methods [27, 31, 40, 41, 44] still follow the conventional video tracking paradigms: they first convert asynchronous event streams into frame-like representations (e.g., dense voxel grid [49] and event count image [26]), then apply appearance-based or motion-based methods for object detection and tracking [12, 20]. While this "events-to-frames" conversion facilitates the use of mature image processing algorithms, it weakens the inherent advantages of event data—namely, its temporal continuity and spatial sparsity. This paradigm faces three critical challenges when tracking fast-moving small objects: 1) **Temporal information loss:** projecting events into frames results in temporal downsampling, disrupting original continuous motion trajectories. 2) **Extreme object characteristic:** small objects inherently lack distinctive appearance cues (e.g., weak texture, shape, and color distribution), making them hard to distinguish and reliably associate using traditional appearance modeling. 3) **Heavy computation burden:** converting events to frames introduces substantial background redundancy, imposing significant computational burdens.

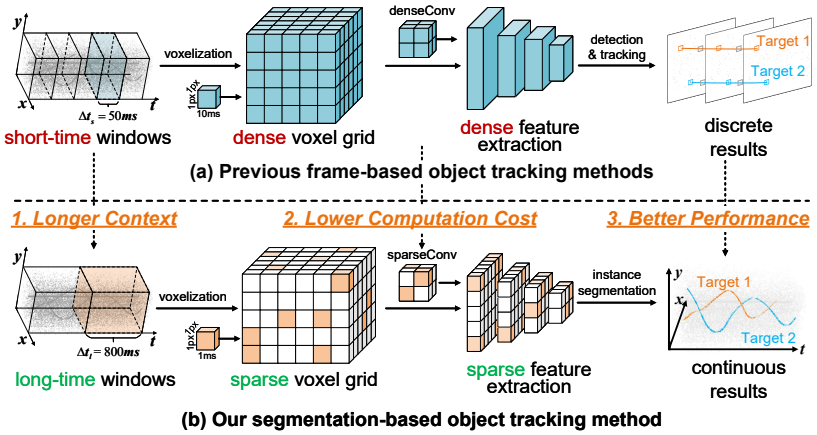

Fig. 2: Comparison of (a) previous methods and (b) our method. Our instance segmentation-based method preserves both the high temporal resolution and inherent sparsity of events. Built upon sparse computation, our method enables parallel processing of a larger number of events and facilitates long-context temporal modeling, enhancing both efficiency and tracking accuracy.

To address these problems, we propose a new paradigm for event-based MSOT, termed "instance segmentation as tracking". Specifically, it reformulates the conventional 2D spatial tracking problem as an instance segmentation task within 3D spatio-temporal event point clouds. Different from the previous "convert-then-detect-and-track" pipeline, this new paradigm operates without sequential detectors and trackers. As a result, it not only avoids the information loss during event-frame conversion, but also preserves the inherent sparsity of event, thereby enabling highly efficient computation, as shown in Fig. 2.

However, directly applying this paradigm is challenging. Existing 3D instance segmentation methods [10, 15, 17] fall short in two aspects: (1) most 3D instance segmentation methods assume instances exhibit compact, centroid-centered distributions. This assumption fails for slender, high-speed event trajectories, as their centroids often lie outside the actual curves. (2) Conventional instance segmentation methods are designed for static data, making them incompatible with the streaming processing requirements of object tracking.

To bridge this gap, we propose Event Instance Segmentation Network (EvISNet), which eliminates the reliance on the centroid-centered distribution assumption and operates in a streaming fashion. Specifically, our network first takes fixed-interval event packets as input, and employs a lightweight 3D sparse U-Net [13, 32] to extract per-event features while jointly predicting foreground masks and motion vectors. Then we construct a candidate event graph using k-nearest neighbors, generating candidate edges between adjacent events. A learnable edge prediction network assesses instance affinity between neighboring nodes based on geometric consistency. Subsequently, we extract connected subgraphs as trajectory segments from the filtered event graph using a union-find algorithm.

Finally, an inter-graph association module merges multiple event packets into complete trajectories, enabling streaming inference. To address the lack of large-scale benchmarks for event-based MSOT, we develop a new event-based tracking dataset (namely EV-UAV-Track) based on the existing object detection dataset EV-UAV [8]. Extensive experiments on the newly developed tracking dataset and Bee Swarm dataset [2] show the superiority of our method as compared to 16 state-of-the-art methods. The main contributions are summarized as follows:

- We propose a novel paradigm that models event-based MSOT as an instance segmentation task within 3D spatio-temporal event point clouds, effectively avoiding both temporal information loss and background redundancy caused by converting events to frame-like representations.
- We propose Ev-ISNet, the first MSOT method within 3D event point clouds. It replaces the centroid-centered assumption with geometric consistency constraints to accurately segment slender trajectories, and employs intra- and inter-packet association mechanisms to achieve continuous, online tracking.
- We propose the first large-scale benchmark dataset for MSOT, termed EV-UAV-Track, featuring fine-grained event-point level tracking annotations. Extensive experiments on this dataset and the Bee Swarm dataset fully validate the effectiveness of our method.

2 Related Work

2.1 Event-based Object Tracking Datasets

While event cameras offer significant advantages for tracking high-speed targets, the development of MSOT has been hindered by a lack of specialized datasets. As summarized in Table 1, existing event-based tracking datasets primarily focus on regular-sized objects or single-object tracking (SOT) scenarios. For instance, EventVOT [38] and VisEvent [37] contain a substantial number of sequences but are limited to SOT tasks. Although the FRED dataset [25] introduces multi-object tracking (MOT) capabilities across 230 sequences, its targets remain large (i.e., 129×98 pixels in average). The Bee Swarm dataset [2] provides a challenging MOT scenario with much smaller objects, but is extremely limited in scale, consisting of only a single sequence. To bridge this gap, we propose EV-UAV-Track, the first large-scale dataset dedicated to MSOT. It features extremely small objects (i.e., 6.8×5.4 pixels in average) and provides a diverse collection of 147 sequences.

Table 1: Comparison of existing event-based tracking datasets. "MOT" indicates multi-object tracking.

Dataset	Object Size	# Sequences	MOT
EventVOT [38]	129×100 pixels	1141	×
VisEvent [37]	47.1×55 pixels	820	×
FRED [25]	129×98 pixels	230	✓
Bee Swarm [2]	8.7×7.9 pixels	1	✓
EV-UAV-Track	6.8×5.4 **pixels**	**147**	✓

2.2 Event-based Object Tracking Methods

Most existing event-based tracking methods [1, 36–38] follow a "detect-then-track" paradigm. They first utilize event detectors [12, 24, 30, 35] for object detection, and subsequently employ existing trackers for tracking [6, 11, 14, 34,

39, 46, 48]. Depending on the tracking mechanisms used, these methods generally fall into two categories: (i) Motion-based tracking methods [6, 39] rely on kinematic models and motion consistency metrics. However, framing continuous events causes severe temporal downsampling, making tracking highly vulnerable to rapid maneuvers and frequent occlusions. (ii) Appearance-based tracking methods [11, 34, 46, 48] use deep networks to extract discriminative features for association. Yet, this struggles in event-based tracking since small targets often appear as textureless sparse points or blobs, severely undermining feature discriminability among similar-sized objects. Although some asynchronous tracking methods [43, 47] have emerged to bypass the framing process, they still exhibit a noticeable performance gap compared to their frame-based counterparts.

To overcome these limitations, our method performs data association directly on 3D event point clouds. By directly segmenting all target events in an event-wise manner, we preserve the complete temporal resolution and leverage the full spatiotemporal context, effectively mitigating the information loss inherent in conventional paradigms.

3 Method

Given an event stream $E = \{e_k = (x_k, y_k, t_k, p_k)\}_{k=1}^{N}$, where (x_k, y_k) denotes the pixel coordinates, t_k is the timestamp of the event trigger, and $p_k \in \{+1, -1\}$ indicates the event polarity (with $+1$ and -1 corresponding to an increase and decrease in brightness, respectively). When a small object moves in the scene, the events it triggers form continuous trajectories in the event spatio-temporal point cloud space. Our goal is to predict all such trajectories and distinguish them, which is analogous to instance segmentation in 3D point clouds, where each trajectory is treated as a distinct instance. Taking the event stream E as input, our method predicts for each event whether it belongs to a target, and assigns it a trajectory label $T^i = \left\{t_j^i\right\}_{j=1}^{N}$, where N is the number of input events and i denotes the instance label of the trajectory.

The overall architecture is illustrated in Fig. 3. The network takes consecutive event packets, split at fixed time intervals, as input. First, an event-wise prediction network (Sec. 3.1) extracts features and predicts per-event semantic labels and direction vectors. Then, the intra-graph association model (Sec. 3.2) aggregates events within each event packet. Finally, the inter-graph association module (Sec. 3.3) aggregates trajectories across adjacent packets and produces the final instance segmentation result.

3.1 Event-wise Prediction

The event-wise prediction network takes the event packet $E_i \in \mathbb{R}^{N\times 4}$ as input, where N is the number of events. The 4 channels contain spatial locations (x, y), timestamp t and polarity p. We first convert the event stream into a sparse voxel grid, which is then fed into a U-Net-like structure composed of

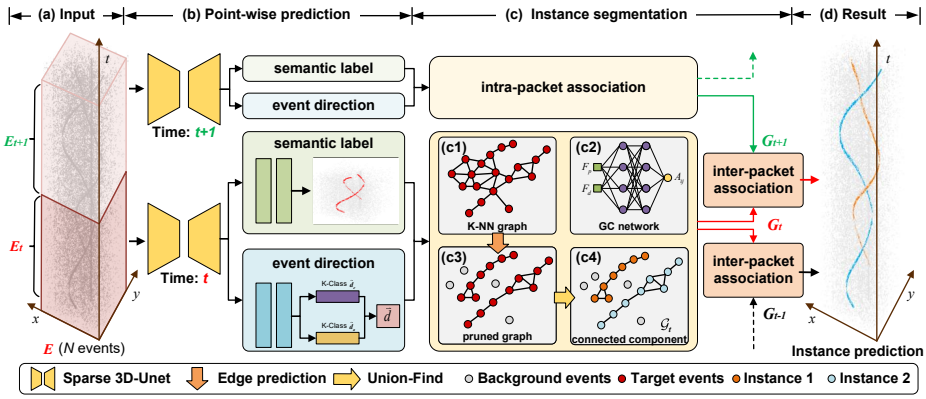

Fig. 3: The framework of our Ev-ISNet. (a) Continuous event streams are partitioned into event packets with fixed time intervals as input. (b) A 3D U-Net architecture with submanifold sparse convolution performs feature learning for each event. The semantic and direction branches predict target probabilities and motion directions, respectively. (c) Target events serve as input for instance segmentation: (c1) events are modeled as a K-Nearest Neighbor (K-NN) graph; (c2) instance affinity between events is predicted via a Geometric Consistency (GC) network; (c3) edges connecting non-identical instances are removed; (c4) connected subgraphs within the graph are computed via a Union-Find algorithm. Finally, the inter-packet association module links trajectories across adjacent event packets. Intra- and inter-packet association modules share similar structures, differing primarily in candidate generation: the former constructs edges within a single event packet, while the latter establishes connections between consecutive packets. (d) The network generates event-wise tracking results.

stacked 3D sparse convolutions to extract voxel features F_{voxel}. These voxel features are subsequently mapped back to event features F_{event}. Based on event features, we construct two branches: one for predicting event-wise semantic labels, and the other for estimating a direction vector for each event. Notably, unlike previous dense voxel representations [49], we voxelize the event stream in a non-overlapping manner at an extremely fine resolution of 1 *pixel* × 1 *pixel* × 1 *ms*, which preserves more temporal information of the events. Further analysis can be found in the experiments section (Sec. 5.3).

Semantic Label Prediction Branch. This is a lightweight binary classification network composed of a two-layer Multi-Layer Perception (MLP). It takes event features F_{event} as input and predicts a target probability for each event. Events with scores above a threshold τ_{object} are classified as targets, while the rest are treated as background. This branch is trained end-to-end using binary cross-entropy (BCE) loss.

Direction Prediction Branch. To explicitly address the challenge of target occlusions, we introduce a direction prediction mechanism. By capturing the local motion continuity, the predicted direction allows us to extrapolate the target's move-

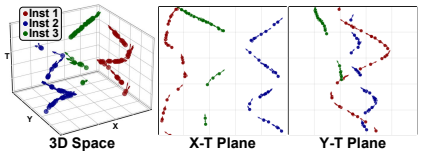

Fig. 4: Geometric consistency of events along the instance trajectory.

ment across temporal blind spots, thereby
providing reliable geometric cues to link
fragmented trajectories. Paralleled with the semantic label prediction branch, we apply a two-layer MLP on the event features F_{event} to predict a per-event direction vector $\boldsymbol{d}$, which indicates the local motion direction of each event. To generate supervision labels for motion direction, we collect the k neighbors of each event and compute its covariance matrix:

$$C = \sum (e_j - \bar{e})(e_j - \bar{e})^T, \tag{1}$$

where $\bar{e} = \frac{1}{k}\sum e_j$ is the geometric center of the neighborhood. The ground-truth motion direction $\boldsymbol{d}_{gt}$ is obtained by normalizing the principal eigenvector of C.

Direct regression of a continuous direction vector is challenging. Therefore, we decompose the 3D direction into azimuth and elevation components and discretize them into a classification task. The direction loss is generated only on target events, formulated as the cross-entropy over the discrete direction classes:

$$\mathcal{L}_{\text{direction}} = \text{CrossEntropy}(M_{obj} \cdot \boldsymbol{d}_{pre}, \boldsymbol{d}_{gt}), \tag{2}$$

where M_{obj} is the mask for target events.

3.2 Intra-packet Association

Existing 3D instance segmentation methods heavily rely on spatial proximity as a prior for grouping points into instances. However, this assumption fails for spatio-temporal event data, where a single small moving object forms a slender, elongated curve that distributed over long ranges. To address this, we propose a graph-based event association method that captures long-range dependencies through bottom-up aggregation, enabling robust segmentation of complete, slender trajectories. Specifically, we first organize the event point cloud into a k-NN graph to establish initial connectivity. We then employ an edge prediction network to estimate the instance affinity between neighboring nodes based on their geometric consistency. Finally, connected subgraphs are extracted as instance segments using a union-find algorithm.

k-NN Graph Construction. We begin by constructing an undirected k-nearest neighbor graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to model connectivity relationships among events. Each node $v_i \in \mathcal{V}$ corresponds to an event e_i, and the edge set $\mathcal{E}$ is obtained by performing k-NN search:

$$\mathcal{E} = \{(i, j) \mid e_j \in \text{kNN}(e_i),\, i \neq j\}. \tag{3}$$

This graph contains $k{\cdot}N$ undirected edges, which serve as the foundation for subsequent connectivity analysis and instance segmentation.

Edge Prediction. We employ a lightweight Geometric Consistency network (GC network) for instance affinity prediction on candidate edges. The network is implemented as a two-layer MLP that takes the feature difference between

connected nodes, i.e., $\Delta F_{ij} = F_i - F_j$, as input and produces a scalar probability:

$$p_{ij} = \sigma(\text{MLP}(\Delta F_{ij})), \quad p_{ij} \in [0, 1], \tag{4}$$

where σ denotes the sigmoid function. Edges with $p_{ij} \geq \tau_{edge}$ are retained for subsequent connected component extraction, thereby enabling instance-aware clustering that does not rely on spatial proximity. As shown in Fig. 4, events exhibit geometric consistency along motion trajectories: neighboring events maintain consistent motion directions.

Instance prediction. We employ a Union-Find data structure to efficiently extract connected components from the pruned graph output by the edge prediction network. Specifically, for each retained edge (i, j) satisfying $p_{ij} \geq \tau$, we execute Union(i, j) to merge all reachable nodes into the same root component. Ultimately, each connected component is identified as a distinct instance.

3.3 Inter-packet Association

Inter-packet association is designed to link identical trajectories across adjacent event packets, thereby enabling streaming object tracking. Its pipeline is similar to that of intra-packet association, with the key difference lying in the candidate edge generation: edges are only established between the two adjacent event packets. Specifically, we first construct cross-packet candidate edges. A lightweight MLP, which shares weights with the intra-packet edge prediction module, is then used to predict their instance affinity scores. Finally, trajectory merging is performed via the union-find algorithm.

Cross-Packet Candidate Edge Generation. Given two adjacent packets $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, we construct a candidate edge set $\mathcal{E}_{12} \subseteq \mathcal{V}_1 \times \mathcal{V}_2$ to connect temporally adjacent nodes from different packets. The candidate set $\mathcal{E}_{12}$ is generated by performing k-NN search and applying a masking operation to retain index pairs where source nodes belong to $\mathcal{V}_1$ and target nodes belong to $\mathcal{V}_2$.

3.4 Multi-task Training

The whole network is trained from scratch in an end-to-end manner and optimized by a joint loss consisting of several terms, as follows:

$$\mathcal{L} = \mathcal{L}_{\text{seg}} + \alpha \mathcal{L}_{\text{direction}} + \beta \mathcal{L}_{\text{edge}}, \tag{5}$$

where $\mathcal{L}_{\text{seg}}$ and $\mathcal{L}_{\text{edge}}$ are the cross-entropy losses of semantic scores and instance affinity, respectively. $\mathcal{L}_{\text{direction}}$ is defined in Eq. (2), while α and β are empirical weights to balance the auxiliary loss terms.

4 EV-UAV-Track Dataset

To facilitate research in event-based MSOT, we introduce the EV-UAV-Track dataset, constructed upon the existing EV-UAV detection dataset. The dataset

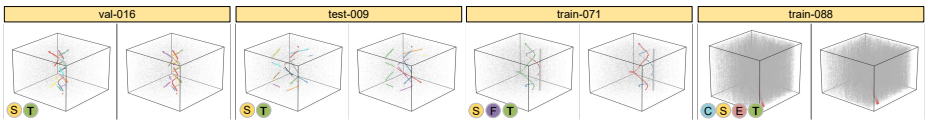

Fig. 5: Examples of our EV-UAV-Track dataset. The left side displays the original EV-UAV dataset, while the right side illustrates our re-annotated data. The top section indicates the sequence name and its assigned partition. Sequence-level attributes are annotated at the bottom, including camera motion (C), extreme illumination (E), scale variation (S), fixed noise (F), and trajectory fragmentation (T).

consists of 147 event sequences containing over 2.3 million target events. The data were captured using a DAVIS346 event camera (346 $\times$ 260 spatial resolution, up to 1 MHz temporal resolution), encompassing various weather and lighting conditions, diverse background scenes, and multiple typical UAV flight maneuvers to ensure broad environmental diversity. While the original EV-UAV dataset provides foundational annotations, its trajectory generation relies on simple inter-frame association, making it susceptible to frequent identity (ID) switches and track fragmentation when handling fast-moving small objects. To overcome these limitations, we comprehensively re-annotate the dataset by assigning fine-grained tracking IDs at the event point level in the 3D spatiotemporal domain. This rigorous re-annotation yields precise, continuous instance-level trajectories, as illustrated in Fig. 5.

5 Experiments

5.1 Implementation Details

Experiment Settings. We partition the event stream into 800-millisecond packets for input and discretize events into a voxel grid with a spatial resolution of 1×1 pixel and a temporal resolution of 1 ms. The batch size is set to 1, and all network layers are initialized with random weights. The model is trained for 50 epochs using the Adam optimizer [16], with a cosine annealing learning rate schedule preceded by a 5-epoch linear warm-up. During warm-up, the learning rate increases linearly from its initial value to a peak of 1×10^{-2}, then decays following a cosine function to a minimum of 1×10^{-3}. The decision thresholds are set to $\tau_{obj} = \tau_{edge} = 0.5$, and the loss weights are set to $\alpha = \beta = 1$. All models are trained on the official training split of the EV-UAV dataset. Model selection is based on the highest performance on the validation set, and the final evaluation is reported on the test set.

Evaluation Metrics. In this work, we evaluate the multi-object tracking performance using MOTA, MOTP and IDF1 metrics [5], while assessing the instance segmentation performance with mCov, mWCov, and mIoU metrics [3]. Since our work focuses on small objects, for which the Intersection-over-Union (IoU) is highly fragile to minor localization errors, we employ the Euclidean distance as the criterion for establishing detection-to-track associations.

Table 2: Quantitative comparison with state-of-the-art methods on the benchmark dataset. *Dense.*, *Sparse.*, *Short.*, *Long.*, and *SNN* denote dense voxel grid, sparse voxel grid, short-term context, long-term context, and spiking neural networks, respectively. *Runtime* is reported for processing an 8-second event sequence. The **bold** and underline represent the best and second-best performance, respectively.

Method	Event Rep.	Multi-Object Tracking			Instance Segmentation			#Params↓ (*MB*)	Runtime ↓ (*second*)
		MOTA ↑	MOTP ↓	IDF1 ↑	mCov ↑	mWCov ↑	mIoU ↑		
CenterTrack [48]	Dense.+Short.	26.80	8.92	16.61	12.22	12.34	13.23	114.1	4.13
FairMOT [46]	Dense.+Short.	31.72	7.35	12.58	11.02	11.52	12.18	247.3	3.53
ByteTrack [45]	Dense.+Short.	29.83	7.91	18.67	8.89	5.74	9.52	68.5	3.12
MOTIP [11]	Dense.+Short.	33.89	8.01	16.28	13.35	11.15	12.36	465.8	6.87
RVT [12]	Dense.+Short.								
+ SORT [6]	Dense.+Short.	37.08	6.23	21.71	22.17	9.15	17.68	9.9	1.92
+ DeepSort [39]	Dense.+Short.	36.15	6.11	22.81	22.24	9.22	18.75	13.0	2.43
+ TrackTrack [34]	Dense.+Short.	36.91	6.99	23.16	22.20	9.19	17.01	319.2	2.24
SpikeYOLO [24]	SNN+Short.								
+ SORT [6]	SNN+Short.	33.82	6.67	22.74	24.05	9.35	16.95	69.0	2.04
+ DeepSort [39]	SNN+Short.	34.88	6.87	23.87	24.12	9.41	17.02	72.1	2.56
+ TrackTrack [34]	SNN+Short.	34.15	6.91	24.68	24.08	9.98	16.98	378.3	2.36
SNNTracker [47]	SNN.+Long.	35.20	5.53	26.11	24.30	10.21	17.19	24.8	3.18
PointGroup [15]	Sparse.+Long.	51.19	2.38	29.86	30.72	11.45	20.18	29.6	1.98
HAIS [10]	Sparse.+Long.	52.71	1.95	30.19	32.35	12.92	22.29	353.2	1.57
OneFormer3D [17]	Sparse.+Long.	55.09	1.81	33.21	31.21	12.19	30.18	220.4	1.87
Ev-ISNet (Ours)	Sparse.+Long.	**64.48**	**1.29**	**48.73**	**37.12**	**18.51**	**36.49**	**7.73**	**1.48**

5.2 Comparison to State-of-the-arts Methods

Results on EV-UAV-Track. We conduct comprehensive comparisons with a range of state-of-the-art methods, including: frame-based generic object detection and tracking methods (i.e., CenterTrack [48], FairMOT [46], ByteTrack [45], MOTIP [11]); event-based object detection methods (i.e., RVT [12] and SpikeYOLO [24]), each integrated with 3 distinct trackers (i.e., SORT [6], DeepSort [39], TrackTrack [34]); dedicated asynchronous event tracker (i.e., SNNTracker [47]); and point cloud-based instance segmentation methods (i.e., PointGroup [15], HAIS [10], OneFormer3D [17]). Following the setting in [12], we convert events within each 50 ms time window into a dense voxel grid representation to adapt to frame-based detectors. Following [8], we convert bounding box-based detection and tracking results into point-based instance segmentation results, and vice versa, to enable comprehensive evaluation of different methods under both frame-based and point-based performance metrics.

Quantitative Results. As demonstrated in Tab. 2, our Ev-ISNet achieves the best tracking performance across all evaluation metrics. The methods based on the "instance segmentation as tracking" paradigm significantly outperform other methods. We attribute this superiority to two key factors: 1) The semantic segmentation-based paradigm preserves finer-grained spatio-temporal information, while the computational efficiency gained from sparse convolutions enables our method to incorporate longer temporal context. This enhances the network's ability to model distinctions between target objects and background noise. 2) The geometry-consistent instance affinity prediction method captures long-range

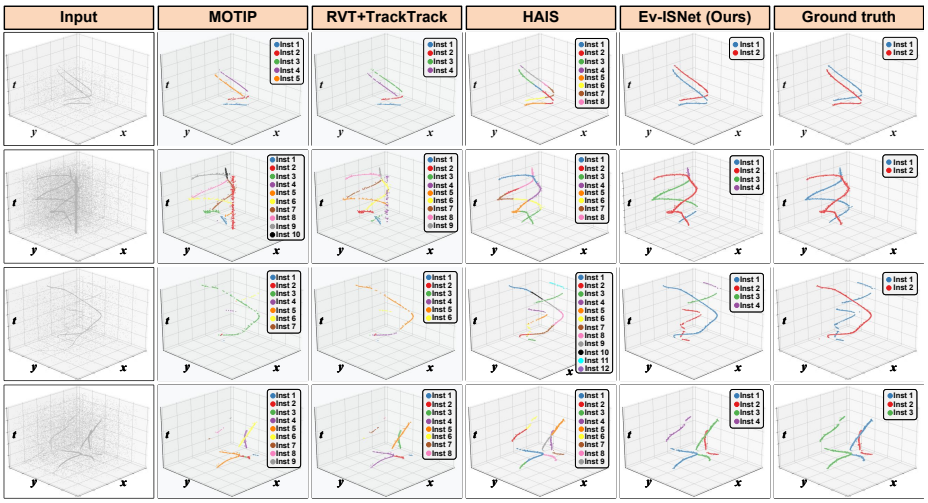

Fig. 6: Qualitative comparison of other methods and Ev-ISNet. From left to right: input events, dense voxel-based methods (i.e., MOTIP [11], RVT [12]+TrackTrack [34]), sparse voxel-based methods (i.e., HAIS [10], Ours), and ground truth. Different colors represent different trajectories (3D instances).

Table 3: Performance comparison on Bee Swarm Dataset. Our Ev-ISNet achieves state-of-the-art performance in dense and highly dynamic tracking scenarios.

Method	MOTA↑	MOTP↓	IDF1↑	mCov↑	mWCov↑	mIOU↑
MOTIP [11]	16.2	4.12	12.4	10.5	8.9	13.8
RVT [12]+SORT [6]	19.3	3.94	15.5	15.1	10.2	13.5
HAIS [10]	23.1	1.55	20.6	20.3	16.5	22.1
SNNTracker [47]	22.1	1.71	18.6	19.2	15.9	20.9
Ev-ISNet (Ours)	**31.5**	**1.13**	**28.9**	**24.1**	**19.8**	**28.3**

dependencies through progressive diffusion in a bottom-up manner, allowing distant events to be associated with the same instance. This provides a more robust relational representation, facilitating the complete extraction of slender curvilinear structures.

Qualitative Results. Qualitative results of different methods are shown in Fig. 6. It can be observed that compared to the conventional "convert-then-detect-and-track" paradigm (i.e., MOTIP [11] and RVT [12]+TrackTrack [34]), our proposed "Instance Segmentation as Tracking" paradigm achieves superior detection performance, capturing significantly more event points while substantially reducing missed detections and false positives prevalent in traditional approaches. When compared to existing 3D instance segmentation methods (i.e., HAIS [10]), our method demonstrates enhanced tracking capability, particularly in maintaining consistency along slender trajectories. This improvement stems from our geometry-consistent association mechanism, which propagates linkages across distant event points through diffusion-based correlation.

Results on Bee Swarm Dataset. The Bee Swarm dataset [2] is a 5-second recording captured by a Prophesee EVK4 HD event camera, comprising approx-

Table 4: Ablation studies on the components. *Dir.*, *Intra.*, and *Inter.* denote direction, intra-packet, and inter-packet association.

Dir.	*Intra.*	*Inter.*	MOTA ↑	MOTP ↓	IDF1 ↑	mCov ↑	mWCov ↑	mIOU ↑	Online
			52.71	1.95	30.19	32.35	12.92	22.29	No
✓			53.89	1.76	32.23	34.84	13.28	24.16	No
	✓		60.84	1.30	46.81	38.80	16.84	34.98	No
	✓	✓	60.85	1.31	46.82	38.82	16.85	34.99	Yes
✓	✓	✓	**64.48**	**1.29**	**48.73**	**37.12**	**18.51**	**36.49**	Yes

imately 5 million manually labeled events. It features an active swarm where hundreds of bees fly in all directions, with an average of 80 labeled bees present in the image plane at arbitrary given time. This dataset introduces significant tracking challenges because the bees occupy only a few pixels and are set against active background noise from wind-blown trees.

To evaluate the generalization of our method in extremely dense and erratic motion scenarios, we conduct additional experiments on this dataset. As shown in Tab. 3, Ev-ISNet significantly outperforms existing methods across all tracking and segmentation metrics. These results demonstrate that our instance segmentation paradigm is highly robust for MSOT.

5.3 Ablation Study

Contribution of Components. The ablation studies in Tab. 4 validate the contribution of each proposed component. Several key observations can be drawn: First, the intra-packet association module serves as the most critical component, yielding substantial improvements across all metrics (e.g., +8.13 MOTA, +16.62 IDF1), which underscores its fundamental role in establishing robust tracklets from continuous events. Second, the addition of the inter-packet association enables online processing with negligible performance drop, effectively bridging the gap between offline and online paradigms. Third, while the direction prediction module alone provides a moderate gain, it delivers complementary benefits when integrated with the association modules, pushing the overall performance to the state-of-the-art. Visual comparisons are shown in Fig. 7.

The Impact of Voxel Temporal Resolution. Since the voxel temporal resolution determines the rate of temporal information loss in event representation, we investigate its impact on tracking performance, as shown in Fig. 8. When the temporal resolution is coarsened from 1 ms to 50 ms per voxel, 10.4% of the events are compressed into the same spatiotemporal bins, leading to a significant loss of temporal distinction. Correspondingly, the tracking accuracy drops substantially, with MOTA decreasing from 64.4% to 45.7%. These results confirm that maintaining fine-grained temporal resolution is crucial for preserving motion details and achieving robust tracking performance.

The Impact of the Geometric Features. We conduct an ablation study to evaluate the contribution of different geometric features for tracking performance. As shown in Tab. 5, both location and direction features contribute

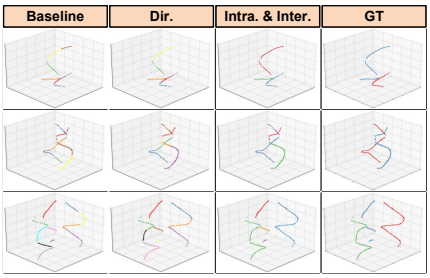

Fig. 7: Qualitative results demonstrating the contribution of the direction prediction and association modules to improved instance segmentation.

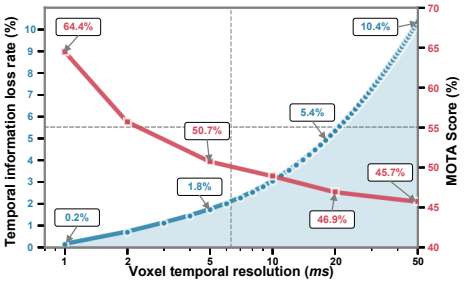

Fig. 8: The impact of the temporal voxel resolution on tracking performance. Accumulating events over longer time intervals causes a significant loss of temporal information, leading to a decline in tracking accuracy.

Table 5: The impact of geometric features on tracking performance.

Location		Direction		MOTA	MOTP	IDF1	mCov	mWCov	mIoU
L	ΔL	D	ΔD	(%)↑	(px)↓	(%)↑	(%)↑	(%)↑	(%)↑
✓				55.73	1.98	42.04	31.45	14.29	26.37
	✓			60.85	1.31	46.82	35.82	16.85	34.99
		✓		42.35	2.56	38.86	20.63	9.82	14.23
			✓	45.23	2.13	40.75	26.63	10.82	17.32
	✓		✓	**64.48**	**1.29**	**48.73**	**37.12**	**18.51**	**36.49**
✓	✓	✓	✓	58.22	1.43	40.23	29.23	12.30	25.24

to performance improvement, since events that are spatially adjacent or exhibit consistent motion patterns are more likely to belong to the same target instance. Notably, the combination of relative location ($\triangle L$) and relative direction ($\triangle D$) features achieves the best performance in terms of all evaluation metrics, significantly outperforming configurations using only individual or absolute features. This indicates that comparative features capturing inter-event relationships better encode discriminative information that distinguishes one trajectory from another, thereby enhancing tracking robustness. However, when all geometric features—both absolute and relative—are incorporated, performance declines, suggesting the presence of redundancy or interference among the features.

The Impact of Time Window Size. We next conduct an ablation study to analyze the relationship between tracking performance and temporal window size. Tab. 6 demonstrates a clear trend: performance improves substantially as the temporal context expands from 100 ms to 800 ms. This indicates that a sufficiently large window is crucial for aggregating enough temporal information to resolve long-term associations and estimate consistent trajectories. However, further increasing the window size beyond this point to 1200 ms leads to slightly degraded performance. We attribute this to the introduction of irrelevant or outdated information from the distant past, which can contaminate the local motion pattern and hinder data association. Therefore, we set $\triangle T$ to 800 ms,

Table 6: The impact of time window size ΔT on tracking performance.

ΔT (ms)	MOTA↑	MOTP↓	IDF1↑	mCov↑	mWCov↑	mIoU↑
100	55.32	2.47	39.15	26.28	12.73	28.85
200	61.75	1.42	45.82	34.91	17.42	35.26
400	62.16	1.35	46.95	35.98	18.03	36.12
800	**64.48**	**1.29**	**48.73**	**37.12**	**18.51**	**36.49**
1200	63.89	1.38	47.25	36.24	17.86	35.67

Table 7: The impact of the number of neighbors K on tracking performance.

K	MOTA↑	MOTP↓	IDF1↑	mCov↑	mWCov↑	mIoU↑
5	58.32	1.52	42.15	33.28	15.73	31.85
10	61.75	1.43	45.82	35.91	17.42	34.26
15	**64.48**	**1.29**	**48.73**	**37.12**	**18.51**	**36.49**
30	60.89	1.47	44.25	34.24	16.86	33.67

which achieves a balance between leveraging informative long-term context and preserving the relevance of immediate motion cues.

The Impact of Number of Neighbors. As shown in Tab. 7, we conduct an ablation study to investigate the impact of the number of neighbors on tracking performance. The results demonstrate that setting the candidate neighbor count K to 15 yields optimal performance in terms of all metrics. This can be attributed to the trade-off between two opposing effects: an insufficient number of neighbors (i.e., an over-small K) results in a limited receptive field that fails to capture sufficient contextual information for robust association, leading to fragmented trajectories. Conversely, an excessively large K incorporates noisy and irrelevant events from distant neighbors, increasing the risk of identity switches and trajectory merges.

5.4 Real-time Inference and Latency

Table 8: Latency analysis under different sliding window strides.

Stride (ms)	20	50	100	400	800
Latency (ms)	18.2	20.3	35.3	100.3	147.6

While our baseline evaluation employs non-overlapping windows for experimental simplicity, our architecture naturally supports an overlapping sliding-window mechanism for practical, streaming deployment. In this mode, the network continuously updates the tracking state by advancing a smaller temporal stride. As shown in Tab. 8, we evaluate the processing latency under various sliding window strides. With a stride of 20 ms, the processing latency is merely 18.3 ms per update. This demonstrates that our method seamlessly handles high-frequency event streams and easily satisfies the requirements for real-time multi-object tracking.

6 Conclusion

In this paper, we introduce a novel "instance segmentation as tracking" paradigm that formulates multi-small-object tracking as a spatiotemporal instance segmentation task. Following this paradigm, we propose Ev-ISNet, which leverages 3D sparse convolutions to extract voxel-wise features while jointly predicting object confidence and motion direction. A graph-based clustering mechanism progressively groups events into object trajectories through instance-aware edge prediction. To support training and evaluation, we present EV-UAV-Track, an

extended version of the EV-UAV dataset enriched with tracking annotations. Experimental results demonstrate that our method consistently outperforms state-of-the-art tracking approaches, achieving over 30% improvement in MOTA compared to existing frame-based methods.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62401590.

References

1. Afshar, S., Nicholson, A.P., Van Schaik, A., Cohen, G.: Event-based object detection and tracking for space situational awareness. IEEE Sensors Journal **20**(24), 15117–15132 (2020)
2. Apps, A., Wang, Z., Perejogin, V., Molloy, T.L., Mahony, R.: Asynchronous multi-object tracking with an event camera. In: IEEE ICRA. pp. 794–800 (2025)
3. Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S.: 3d semantic parsing of large-scale indoor spaces. In: CVPR. pp. 1534–1543 (2016)
4. Baldwin, R.W., Liu, R., Almatrafi, M., Asari, V., Hirakawa, K.: Time-ordered recent event (tore) volumes for event cameras. IEEE TPAMI **45**(2), 2519–2532 (2022)
5. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. EURASIP Journal on Image and Video Processing **2008**(1), 246309 (2008)
6. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: ICIP. pp. 3464–3468 (2016)
7. Chen, N., Li, B., Wang, Y., Ying, X., Wang, L., Zhang, C., Guo, Y., Li, M., An, W.: Motion and appearance decoupling representation for event cameras. IEEE TIP **34**, 5964–5977 (2025)
8. Chen, N., Xiao, C., Dai, Y., He, S., Li, M., An, W.: Event-based tiny object detection: A benchmark dataset and baseline. In: ICCV. pp. 7209–7218 (October 2025)
9. Chen, N., Zhang, C., An, W., Wang, L., Li, M., Ling, Q.: Event-based motion deblurring with blur-aware reconstruction filter. IEEE TCSVT **35**(9), 8508–8519 (2025)
10. Chen, S., Fang, J., Zhang, Q., Liu, W., Wang, X.: Hierarchical aggregation for 3d instance segmentation. In: ICCV. pp. 15467–15476 (2021)
11. Gao, R., Qi, J., Wang, L.: Multiple object tracking as id prediction. In: CVPR. pp. 27883–27893 (2025)
12. Gehrig, M., Scaramuzza, D.: Recurrent vision transformers for object detection with event cameras. In: CVPR. pp. 13884–13893 (2023)
13. Graham, B., Engelcke, M., Van Der Maaten, L.: 3d semantic segmentation with submanifold sparse convolutional networks. In: CVPR. pp. 9224–9232 (2018)
14. Hong, L., Yan, S., Zhang, R., Li, W., Zhou, X., Guo, P., Jiang, K., Chen, Y., Li, J., Chen, Z., et al.: Onetracker: Unifying visual object tracking with foundation models and efficient tuning. In: CVPR. pp. 19079–19091 (2024)

15. Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C.W., Jia, J.: Pointgroup: Dual-set point grouping for 3d instance segmentation. In: CVPR. pp. 4867–4876 (2020)
16. Kingma, D.P., Ba, J.L.: Adam: Amethod for stochastic optimization. In: ICLR. pp. 1–15 (2014)
17. Kolodiazhnyi, M., Vorontsova, A., Konushin, A., Rukhovich, D.: Oneformer3d: One transformer for unified point cloud segmentation. In: CVPR. pp. 20943–20953 (2024)
18. Li, D., Tian, Y., Li, J.: Sodformer: Streaming object detection with transformer using events and frames. IEEE TPAMI **45**(11), 14020–14037 (2023)
19. Li, J., Dong, S., Yu, Z., Tian, Y., Huang, T.: Event-based vision enhanced: A joint detection framework in autonomous driving. In: ICME. pp. 1396–1401. IEEE (2019)
20. Li, J., Li, J., Zhu, L., Xiang, X., Huang, T., Tian, Y.: Asynchronous spatio-temporal memory network for continuous event-based object detection. IEEE TIP **31**, 2975–2987 (2022)
21. Li, J., Tian, Y.: Recent advances in neuromorphic vision sensors: A survey. Chinese Journal of Computers **44**(6), 1258–1286 (2021)
22. Li, J., Wang, X., Zhu, L., Li, J., Huang, T., Tian, Y.: Retinomorphic object detection in asynchronous visual streams. In: AAAI. vol. 36, pp. 1332–1340 (2022)
23. Liang, Q., Lu, J., Li, Q., Liu, S., Zhao, Z., Zhao, Y., Zhang, W., Huang, K., Tian, Y.: Esod: Event-based small object detection. In: ACM MM. pp. 518–527 (2025)
24. Luo, X., Yao, M., Chou, Y., Xu, B., Li, G.: Integer-valued training and spike-driven inference spiking neural network for high-performance and energy-efficient object detection. In: ECCV. pp. 253–272. Springer (2024)
25. Magrini, G., Marini, N., Becattini, F., Berlincioni, L., Biondi, N., Pala, P., Del Bimbo, A.: Fred: The florence rgb-event drone dataset. In: ACM MM. pp. 13170–13176 (2025)
26. Maqueda, A.I., Loquercio, A., Gallego, G., García, N., Scaramuzza, D.: Event-based vision meets deep learning on steering prediction for self-driving cars. In: CVPR. pp. 5419–5427 (2018)
27. Mitrokhin, A., Fermüller, C., Parameshwara, C., Aloimonos, Y.: Event-based moving object detection and tracking. In: IEEE IROS. pp. 1–9 (2018)
28. Peng, Y., Li, H., Zhang, Y., Sun, X., Wu, F.: Scene adaptive sparse transformer for event-based object detection. In: CVPR. pp. 16794–16804 (2024)
29. Peng, Y., Zhang, Y., Xiong, Z., Sun, X., Wu, F.: Get: Group event transformer for event-based vision. In: ICCV. pp. 6038–6048 (2023)
30. Perot, E., De Tournemire, P., Nitti, D., Masci, J., Sironi, A.: Learning to detect objects with a 1 megapixel event camera. NeurIPS **33**, 16639–16652 (2020)
31. Ramesh, B., Zhang, S., Lee, Z.W., Gao, Z., Orchard, G., Xiang, C.: Long-term object tracking with a moving event camera. In: BMVC. p. 241 (2018)
32. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. pp. 234–241 (2015)
33. Shi, Y., Li, M., Chen, N., Luo, Y., He, S., An, W.: Sparse-gated rgb-event fusion for small object detection in the wild. Remote Sensing **17**(17), 3112 (2025)
34. Shim, K., Ko, K., Yang, Y., Kim, C.: Focusing on tracks for online multi-object tracking. In: CVPR. pp. 11687–11696 (2025)
35. Su, Q., Chou, Y., Hu, Y., Li, J., Mei, S., Zhang, Z., Li, G.: Deep directly-trained spiking neural networks for object detection. In: ICCV. pp. 6555–6565 (2023)
36. Tang, C., Wang, X., Huang, J., Jiang, B., Zhu, L., Chen, S., Zhang, J., Wang, Y., Tian, Y.: Revisiting color-event based tracking: A unified network, dataset, and metric. PR p. 112718 (2025)

37. Wang, X., Li, J., Zhu, L., Zhang, Z., Chen, Z., Li, X., Wang, Y., Tian, Y., Wu, F.: Visevent: Reliable object tracking via collaboration of frame and event flows. IEEE TCYB **54**(3), 1997–2010 (2023)
38. Wang, X., Wang, S., Tang, C., Zhu, L., Jiang, B., Tian, Y., Tang, J.: Event stream-based visual object tracking: A high-resolution benchmark dataset and a novel baseline. In: CVPR. pp. 19248–19257 (2024)
39. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: ICIP. pp. 3645–3649 (2017)
40. Wu, Z., Zheng, J., Ren, X., Vasluianu, F.A., Ma, C., Paudel, D.P., Van Gool, L., Timofte, R.: Single-model and any-modality for video object tracking. In: CVPR. pp. 19156–19166 (2024)
41. Yen, T.K., Morawski, I., Dangi, S., He, K., Lin, C.Y., Yeh, J.F., Su, H.T., Hsu, W.: Tracking-assisted object detection with event cameras. In: ECCV. pp. 138–155 (2024)
42. Ying, X., Xiao, C., An, W., Li, R., He, X., Li, B., Cao, X., Li, Z., Wang, Y., Hu, M., Xu, Q., Lin, Z., Li, M., Zhou, S., Liu, L., Sheng, W.: Visible-thermal tiny object detection: A benchmark dataset and baselines. IEEE TPAMI **47**(7), 6088–6096 (2025)
43. Zhang, J., Dong, B., Zhang, H., Ding, J., Heide, F., Yin, B., Yang, X.: Spiking transformers for event-based single object tracking. In: CVPR. pp. 8801–8810 (2022)
44. Zhang, J., Yang, X., Fu, Y., Wei, X., Yin, B., Dong, B.: Object tracking by jointly exploiting frame and event domain. In: ICCV. pp. 13043–13052 (2021)
45. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. In: ECCV. pp. 1–21 (2022)
46. Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W.: Fairmot: On the fairness of detection and re-identification in multiple object tracking. IJCV **129**(11), 3069–3087 (2021)
47. Zheng, Y., Li, C., Zhang, J., Yu, Z., Huang, T.: Snntracker: Online high-speed multi-object tracking with spike camera. IEEE TPAMI (2025)
48. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. In: ECCV. pp. 474–490 (2020)
49. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Unsupervised event-based learning of optical flow, depth, and egomotion. In: CVPR. pp. 989–997 (2019)
50. Zhu, Y., Li, C., Liu, Y., Wang, X., Tang, J., Luo, B., Huang, Z.: Tiny object tracking: A large-scale dataset and a baseline. IEEE TNNLS **35**(8), 10273–10287 (2023)