Rethinking Scale-Aware Temporal Encoding for Event-based Object Detection

Lin Zhu[1], **Tengyu Long**[1], **Xiao Wang**[2], **Lizhi Wang**[3], **Hua Huang**[3,*]
[1] School of Computer Science, Beijing Institute of Technology
[2] School of Computer Science and Technology, Anhui University
[3] School of Artificial Intelligence, Beijing Normal University
linzhu@bit.edu.cn, longtengyu@bit.edu.cn
xiaowang@ahu.edu.cn, wanglizhi@bnu.edu.cn, huahuang@bnu.edu.cn

Abstract

Event cameras provide asynchronous, low-latency, and high-dynamic-range visual signals, making them ideal for real-time perception tasks such as object detection. However, effectively modeling the temporal dynamics of event streams remains a core challenge. Most existing methods follow frame-based detection paradigms, applying temporal modules only at high-level features, which limits early-stage temporal modeling. Transformer-based approaches introduce global attention to capture long-range dependencies, but often add unnecessary complexity and overlook fine-grained temporal cues. In this paper, we propose a CNN-RNN hybrid framework that rethinks temporal modeling for event-based object detection. Our approach is based on two key insights: (1) introducing recurrent modules at lower spatial scales to preserve detailed temporal information where events are most dense, and (2) utilizing Decoupled Deformable-enhanced Recurrent Layers specifically designed according to the inherent motion characteristics of event cameras to extract multiple spatiotemporal features, and performing independent downsampling at multiple spatiotemporal scales to enable flexible, scale-aware representation learning. These multi-scale features are then fused via a feature pyramid network to produce robust detection outputs. Experiments on Gen1, 1 Mpx and eTram dataset demonstrate that our approach achieves superior accuracy over recent transformer-based models, highlighting the importance of precise temporal feature extraction in early stages. This work offers a new perspective on designing architectures for event-driven vision beyond attention-centric paradigms.

Code: `https://github.com/BIT-Vision/SATE`

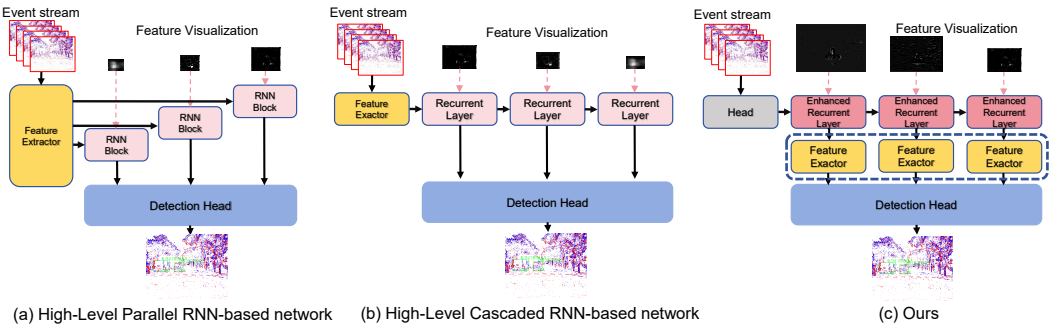
1 Introduction

Event cameras provide a fundamentally different visual sensing modality by asynchronously recording pixel-level brightness changes with microsecond latency [8]. Their high dynamic range, sparse output, and low power consumption make them well-suited for real-time perception in high-speed or low-light environments [46]. These characteristics render event cameras particularly attractive for performing visual tasks such as object detection [27, 31], tracking [9, 39], and optical flow estimation [44, 6, 42], even in challenging scenarios (e.g., extreme lighting variations and high-speed motion dynamics).

Event data is stored as asynchronous arrays containing the spatial coordinates, polarity, and timestamp of each illumination change. In contrast, conventional frame-based cameras represent visual information through fixed-rate pixel value matrices. This fundamental discrepancy renders conventional

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) High-Level Parallel RNN-based network (b) High-Level Cascaded RNN-based network (c) Ours

Figure 1: Comparison of different temporal modeling strategies in event-based object detection. Specifically, (a) adopts a *High-Level Parallel RNN-based network* (e.g., DMANet [38]), where RNN modules are independently inserted at different feature scales. (b) shows a *High-Level Cascaded RNN-based network* (e.g., RED[31], RVT [11], SAST [29]), in which RNN modules are stacked sequentially across multiple scales to achieve cascaded temporal aggregation. In contrast, (c) illustrates *Ours*, where enhanced recurrent modules are inserted *before significant downsampling*. Multi-scale features are extracted separately via scale-specific branches. Here, RNN Block refers to basic recurrent units such as ConvLSTM [33], while Recurrent Layer combines recurrent modeling with additional feature extraction operations. The proposed Enhanced Recurrent Layer design is detailed in Sec. 3.

image-based neural networks incompatible with event data processing. To address this issue, existing methodologies are primarily categorized into two methodological branches. The first approach involves transforming raw spatiotemporal event streams into dense representations analogous to multi-channel images (such as voxel grid [45], event histogram [28], and time surface [20, 34]). Such transformations facilitate the application of established computer vision techniques originally designed for frame-based data. The second category adopts sparse computational paradigms, typically employing spiking neural networks (SNNs) or graph-based architectures, yet these frequently encounter challenges including hardware incompatibility and suboptimal accuracy. In this work, we utilize dense representations for their advantages in effectiveness.

In addition, how to effectively modeling the temporal dynamics of sparse event streams remains an open challenge, particularly for dense prediction tasks such as object detection. Recent approaches for event-based object detection span a diverse range of architectures. Transformer-based methods [11, 30] leverage global attention to capture long-range temporal dependencies but are often computationally expensive and less effective at modeling localized, high-frequency temporal patterns. Spiking neural networks (SNNs) [4, 41, 47] exploit the asynchronous nature of events but suffer from limited representation capacity and optimization difficulty. More recently, state-space models [50, 40] such as Mamba have shown potential for sequential modeling but remain underexplored in the context of low-level event representation. Notably, many of these methods emphasize temporal modeling at deeper layers (Figure 1), where spatial resolution is low and temporal cues may already be degraded.

In this work, we argue that accurate modeling of early-stage temporal features is essential for effective event-based object detection. Since events are inherently sparse and localized in both space and time, the most informative temporal structures often appear at low-level representations. Motivated by this, we propose to revisit convolutional architectures and augment them with effective recurrent modules at early stages to explicitly preserve fine-grained temporal information. Our design is not aimed at merely reducing model complexity, but at improving temporal fidelity in the feature extraction process.

To this end, we introduce a CNN-RNN hybrid network tailored for event-based object detection. The network consists of three key components. First, we place recurrent blocks at lower spatial scales to enhance temporal modeling at early stages, enabling the network to capture fine-grained temporal patterns that are critical for understanding sparse event dynamics. Second, we propose a divide-and-conquer methodology that enhances current event features through decoupled per-pixel motion estimation and spatiotemporal feature fusion. Third, we design a multi-branch backbone with independent temporal downsampling at three resolution levels, allowing flexible and complementary spatiotemporal feature extraction across varying receptive fields. The multiple spatiotemporal features extracted by the backbone then are hierarchically propagated to a detection head incorporating a

Feature Pyramid Network [24], leveraging multi-scale features for accurate and temporally consistent object predictions. Despite its architectural simplicity, our model achieves state-of-the-art performance on the Gen1 dataset, outperforming recent transformer- and SNN-based baselines.

In summary, the contributions of this work are:

(1) We propose a CNN-RNN hybrid architecture for event-based object detection that rethinks temporal modeling by introducing recurrent modules at lower spatial scales, enabling the capture of fine-grained temporal dynamics from sparse event streams where information is most dense.

(2) We propose a divide-and-conquer methodology that enhances current event features through strategic utilization of temporal information, that decouples the estimation of per-pixel motion from the feature fusion. Meanwhile, we design a scale-specific spatiotemporal encoding strategy that performs independent temporal downsampling across three resolution branches, allowing the network to flexibly extract complementary multi-scale features, which are later fused via a Feature Pyramid Network for robust detection.

(3) Extensive experiments on the Gen1, 1Mpx and eTram benchmark demonstrate that our approach achieves state-of-the-art performance, outperforming recent transformer-based and SNN-based methods, validating the effectiveness of early-stage temporal modeling.

2 Related Works

Existing event-based object detection methodologies can be broadly categorized into two major branches: sparsity-aware architectures and dense feedforward networks.

Sparsity-aware architectures leverage bio-inspired neural designs, notably graph neural networks (GNNs) and spiking neural networks (SNNs), to directly process asynchronous event streams. GNN-based approaches dynamically construct spatiotemporal graphs through event subsampling strategies [32, 10], identifying temporally and spatially proximate nodes and establishing adaptive edge connections. A key challenge is designing graph topologies that enable efficient information propagation across long spatiotemporal ranges while remaining computationally tractable. SNNs, on the other hand, exploit sparse, time-dependent spike-based computation [4, 1], inherently aligning with the event-driven nature of the data. Like RNNs, SNN neurons maintain temporal states, but emit spikes only when membrane potentials exceed a threshold, introducing non-differentiability into the network. While surrogate gradients [26] can circumvent this issue, they often compromise the sparsity advantage. Despite their theoretical suitability, both GNN- and SNN-based methods face practical limitations in terms of hardware dependency and inferior detection accuracy compared to dense alternatives.

Dense feedforward architectures represent the second major branch. Early methods convert event streams into fixed-duration frame representations and apply standard image-based detectors to each temporal slice independently [3, 16, 2, 18, 15]. These methods largely ignore temporal continuity and motion cues, making them less effective in cases involving occlusion, low texture, or slow motion. Later approaches integrate CNNs with RNNs [31, 22, 38], combining the spatial efficiency of CNNs with the sequential modeling capability of RNNs, yielding significant performance gains. With the advent of the Vision Transformer (ViT)[7], transformer-based architectures have gained prominence in event-based detection as well, leveraging self-attention to model long-range dependencies. Specifically, RVT [11] employs a recurrent vision transformer that integrates local–global self-attention with lightweight LSTM [13]-based temporal aggregation for efficient spatiotemporal modeling, while GET [30] introduces grouped token mechanisms and dual self-attention to jointly capture spatial, temporal, and polarity cues for enhanced event representation. In contrast, SSM [50] replaces recurrent operations with continuous-time state-space modeling[12, 35], enabling efficient and frequency-robust temporal aggregation across varying inference rates. While effective, transformers often suffer from computational inefficiency due to the sparsity and noise in event data. Efforts such as SAST [29] attempt to mitigate this by pruning irrelevant tokens and windows via attention-based mechanisms. However, these strategies inevitably compromise the model's global receptive field, limiting their effectiveness in complex scenes.

Our investigation reveals that existing CNN- and Transformer-based approaches combined with recurrent modules primarily emphasize high-level temporal modeling. However, due to the inherent sparsity of event data, reducing spatial resolution often results in the loss of crucial details essential

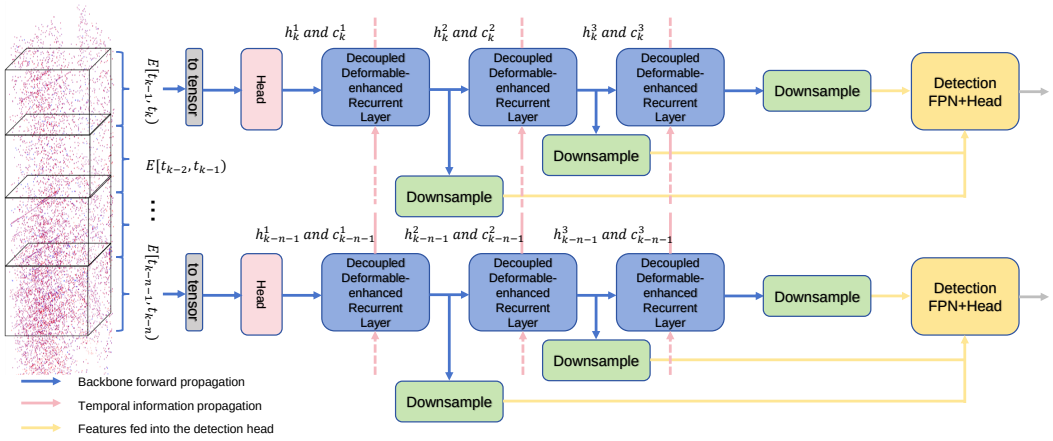


Figure 2: Overview of our designed CNN-RNN network. The event data is first converted into a tensor representation before being fed into the first stage. After passing through a shared shallow stem, the data is processed by our Decoupled Deformable-enhanced Recurrent Layer for spatiotemporal feature extraction. The extracted spatiotemporal features are then further processed in a more flexible manner. Finally, the resulting multi-scale features are fed into the detection head.

for accurate detection. To address this limitation, we strengthen early-stage temporal modeling by integrating event-specific recurrent modules at lower spatial scales, where event activity is denser and temporal cues are richer. Moreover, we introduce a multi-branch downsampling scheme to refine multi-scale spatiotemporal representations, enabling flexible retention of critical information while mitigating information degradation.

3 Method

This section presents the proposed CNN-RNN hybrid architecture for event-based object detection. Our design is motivated by two core observations: (1) fine-grained temporal patterns in event data often appear at early stages and are best captured before spatial abstraction; and (2) scale-specific spatiotemporal encoding enables complementary feature extraction across object sizes and motion dynamics. Meanwhile, we adopt a divide-and-conquer approach to decouple feature fusion from motion estimation, further leveraging temporal information, which not only enhances the current frame features, but also leverages the smoothing effect of deformable convolution to effectively suppress the propagation of task-irrelevant information in low-dimensional features. The overall framework is shown in Figure 2.

3.1 Event Representation

Event cameras output a stream of asynchronous events represented as tuples $e = (x, y, t, p)$, where (x, y) denotes the spatial location, t is the timestamp, and $p \in \{+1, -1\}$ is the polarity indicating intensity increase or decrease. To make the data compatible with convolutional processing, we discretize the event stream into voxel grids [45] over a short temporal window. The events within a temporal window ΔT are converted into a voxel grid of size $B \times H \times W$, where H and W denote the height and width of the event frame, respectively, and B represents the number of temporal bins.

$$V(x, y, t) = \sum_i p_i \delta_b(x - x_i)\delta_b(y - y_i)\delta_b(t - t_i^*), \tag{1}$$

where $t_i^* = \frac{B-1}{\Delta T}(t_i - t_1)$ and $\delta_b(a) = max(0, 1 - |a|)$. Our models use a time window $\Delta T = 50ms$ and $B = 5$ temporal bins.

3.2 Overall Architecture

The proposed model consists of three major components: Early-stage recurrent encoding for fine-grained temporal modeling, Decoupled Deformable-enhanced Recurrent Layer (DDRL) designed

based on the divide-and-conquer principle and the inherent motion information of event data, Scale-specific spatiotemporal branches with independent downsampling.

As shown in Figure 2, the input voxel slice is processed through a shared shallow stem ($stride = 1$) followed by three successive Decoupled Deformable-enhanced Recurrent Layers with built-in downsampling, yielding spatiotemporal features at three different scales. These features are then fed into three parallel branches, each employing its own temporal encoding mechanism at a specific spatial scale. It is worth noting that each $2\times$ downsampling is accompanied by a doubling of the channel dimensions, except for the final downsampling in each branch, which is intended to reduce the overall model complexity. The outputs of all branches are subsequently passed to the detection head for multi-scale object detection.

3.3 Temporal Modeling at Lower Scales

A key challenge in object detection with event cameras lies in whether neural networks can effectively extract meaningful information from both recent events and those generated several seconds earlier. Since event cameras respond to intensity changes caused by object motion, they produce very few events when objects move slowly or remain stationary. This inherent sparsity of event data means that critical information is often embedded in subtle details, which becomes particularly crucial when performing detection across multiple consecutive frames. To preserve this critical temporal information, we place the proposed Decoupled Deformable-enhanced Recurrent Block (DDRB, detailed in Section 3.4) at higher-resolution stages with downsampling factors of 2, 4, and 8. To effectively capture both short-term and long-term dependencies, our model also processes temporal sequences at multiple spatial resolutions. However, unlike previous works [31, 38, 11] emphasizing high-level temporal information propagation, we introduce temporal recurrence *before significant spatial downsampling*, allowing the model to capture dense temporal structures with minimal loss.

In general, the hidden state H_t^s and the cell state C_t^s in recurrent blocks at time t and scale s are updated as:

$$F_t^s = Relu(BN(Conv5 \times 5(F_t^{s-1}))), \tag{2}$$

$$H_t^s, C_t^s = \mathrm{DDRB}(F_t^s, H_{t-1}^s, C_{t-1}^s), \tag{3}$$

where BN means batch normalization [17]. F_t^{s-1} denotes the feature map obtained from the preceding stage, and DDRB represents our proposed Decoupled Deformable-enhanced Recurrent Block, which encapsulates our novel integration of deformable spatial adaptation and temporal recurrence mechanisms.

3.4 Decoupled Deformable-enhanced Recurrent Layer

In event-based object detection tasks, event cameras exclusively detect moving objects, leading to suboptimal performance when relying solely on isolated event frames in scenarios involving slow-moving or stationary targets. ConvLSTM [33], a neural network architecture specifically designed for temporal data sequences, addresses this limitation through its capacity for modeling long-range temporal dependencies. This characteristic enables enhanced detection of objects with low-motion or static states within dynamic scenes. Furthermore, the inherent integration of convolutional operations in ConvLSTM facilitates effective processing of spatial features within visual data. Particularly in event-driven object detection, where distinct spatial regions may contain heterogeneous motion patterns, ConvLSTM assists the model in capturing these spatially distributed temporal characteristics, thereby substantially improving detection capabilities. In our proposed Deformable Enhanced Recurrent Layer, we also employ ConvLSTM for spatiotemporal modeling. However, diverging from conventional implementations, we identify that conventional ConvLSTM structures may underutilize extracted spatiotemporal features. To address this, we propose an innovative fusion module based on modulated deformable convolution [48], specifically designed for the characteristics of event data, by adopting a divide-and-conquer strategy. This architectural enhancement enables comprehensive exploitation of spatiotemporal information generated by ConvLSTM, particularly optimizing feature representation through adaptive receptive field adjustment and motion pattern alignment in complex scenarios. Meanwhile, since the recurrent layers are placed at low-dimensional stages, the low-dimensional features may contain task-irrelevant information in addition to the key information we need. The smoothing effect of the deformable convolution can effectively suppress such redundant information.

Specifically, the Decoupled Deformable-enhanced Recurrent Layer (DDRL) comprises a $2\times$ downsampling module and a Decoupled Deformable-enhanced Recurrent Block (DDRB) (as shown in Figure 3). The downsampling module integrates a convolutional layer with kernel size 5 and stride 2, followed by batch normalization (BN) and ReLU activation, to perform preliminary spatial reduction. The DDRB includes a recurrent module (ConvLSTM), and a decoupled modulated deformable convolution specifically designed based on the characteristics of event data. ConvLSTM is employed to capture spatiotemporal information from consecutive event frames, while the decoupled modulated deformable convolution is used to fuse current frame features with historical spatiotemporal representations. Given the high temporal resolution of event cameras, event data inherently contain fine-grained motion cues. Inspired by this, our design decouples per-pixel motion estimation from feature fusion. We leverage the intrinsic motion information in event data to learn the offsets Δx_k and modulated scalars Δm_k that 1) adapt the sampling grid of the modulated deformable convolution to pixel-wise motion patterns in the scene, and 2) dynamically weight each sampling positions based on their actual contribution to feature representation. The estimated offsets Δx_k and modulation masks Δm_k are applied to the features obtained from the preliminary fusion of the current frame and previous spatiotemporal representations, where the fusion is performed by concatenating along the channel dimension followed by a 3×3 convolution for dimensionality reduction. The deformable convolution subsequently adaptively samples features from the compressed representation using the learned Δx_k to spatially align with motion patterns, while Δm_k dynamically recalibrates sampling weights according to motion relevance. This hierarchical fusion mechanism effectively preserves fine-grained motion cues while eliminating feature redundancy, ensuring optimal utilization of event-driven spatiotemporal correlations.

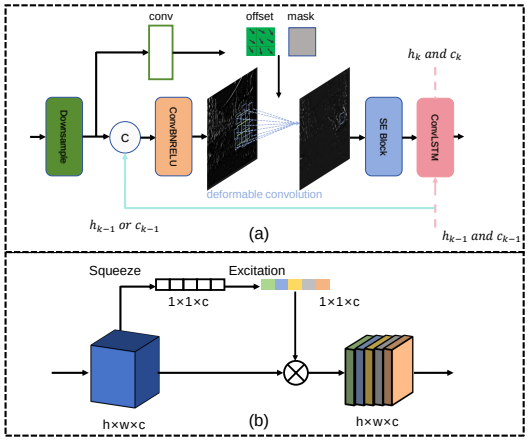


Figure 3: (a) Architecture of the Decoupled Deformable-enhanced Recurrent Layer. Approaches for feature fusion typically employ direct combination through element-wise summation or channel-wise concatenation followed by convolutional integration, or alternatively leverage attention mechanisms for adaptive feature aggregation. Here, we leverage the divide-and-conquer principle to introduce modulated deformable convolution into feature fusion, further exploiting temporal information. (b) Architecture of the SE Block.

Given a convolution kernel with K sampling locations, let w_k and x_k denote the weight and predefined offset of the $k-th$ location, respectively. In this work, we adopt 3×3 convolution kernel, i.e., $K = 9$ and $x_k \in \{(\text{-}1, \text{-}1), (\text{-}1, 0), \cdots, (1, 1)\}$. At time t and scale s, the Decoupled Deformable-enhanced Recurrent Block (DDRB) can be represented as:

$$F_t^{s'} = Relu(BN(Conv3 \times 3(Concat(F_t^s, T_{t-1}^s)))), \tag{4}$$

$$F_t^{s''}(x) = \sum_{k=1}^{K} w_k \cdot F_t^{s'}(x + x_k + \Delta x_k) \cdot \Delta m_k, \tag{5}$$

$$H_t^s, C_t^s = ConvLSTM(SE(F_t^{s''}), H_{t-1}^s, C_{t-1}^s), \tag{6}$$

where Δx_k and Δm_k represent the learnable offset and modulation scalar for the $k-th$ location, respectively, both obtained from the current event feature F_t^s through two separate 3×3 convolution layers. T_{t-1}^s denotes the spatiotemporal information from the previous time step $t-1$ (H_{t-1}^s or C_{t-1}^s). In the experimental section, we investigate the performance differences when using the hidden state H_{t-1}^s, which represents the features of the previous time step, versus the memory cell C_{t-1}^s, which stores long-term spatiotemporal information. In addition, to enable the model to flexibly learn diverse deformation patterns across different spatial regions and enhance its representation capability for complex geometric transformations, we adopt grouped deformable convolutions. Specifically, the input channels are divided into multiple groups (set to 8), with each group independently learning distinct deformation patterns. Furthermore, considering that the offset features generated by different groups may contribute unequally to the final task, we incorporate a Squeeze-and-Excitation (SE)

Block [14] after the grouped deformable convolution. The SE Block[14] leverages global average pooling and fully connected layers to learn channel-wise attention weights, thereby enhancing informative channels and suppressing redundant ones, leading to optimized feature representations.

3.5 Scale-Specific Spatiotemporal Encoding

Multi-scale feature representation has been extensively employed in object detection frameworks to address scale variation challenges, particularly in detecting objects with diverse sizes. For instance, lower-scale branches preserve higher spatial resolution with shallower network depth, making them particularly suitable for detecting small or rapidly moving objects. As feature resolution progressively decreases through the network hierarchy, the expanded receptive fields enable context-aware detection of larger objects. This multi-scale temporal encoding structure allows the model to learn both fine-grained and coarse-grained motion features. In the present work, we process three low-dimensional multi-scale spatiotemporal features derived from preceding layers using three independent processing branches. Each branch implements scale-specific downsampling operations, followed by feature fusion through a Feature Pyramid Network [24]. This hierarchical architecture enables adaptive resolution adjustment across different network levels, effectively preserving critical information with enhanced flexibility while mitigating information degradation during downsampling.

4 Experiment

Dataset. The Gen1 automotive dataset [5] consists of 39 hours of event camera recordings with a resolution of 304×240. It includes $228k$ annotated bounding boxes for vehicles and $28k$ for pedestrians, with available annotation frequencies of 1, 2, or 4 Hz. Following the evaluation protocol of previous works [31, 22, 11], we discard bounding boxes with side lengths smaller than 10 pixels and diagonal lengths shorter than 30 pixels. Similarly, the 1 Mpx dataset [31] focuses on driving scenarios but provides several months of higher-resolution (1280×720) daytime and nighttime recordings. It contains approximately 15 hours of event data, annotated at 30 or 60 Hz, with around 25 million bounding box labels distributed across three categories: vehicles, pedestrians, and two-wheelers. We adhere to the same evaluation protocol, removing bounding boxes with side lengths smaller than 20 pixels and diagonal lengths shorter than 60 pixels, and downsample the input resolution to 640×360. Unlike the Gen1 [5] and 1 Mpx [31] datasets, the eTram dataset [37] is a traffic monitoring dataset collected from a roadside perspective, thus exhibiting higher sparsity. eTram contains approximately 10 hours of data with a resolution of 1280×720, including around 2 million annotated bounding boxes across 8 categories, with annotations provided at 30 Hz. The preprocessing procedure of the eTram dataset is similar to that of the 1 Mpx dataset. For all datasets, mean Average Precision (mAP) [23] is considered as the primary metric.

Implementation Details. During training, we adopt the ADAM optimizer [19] along with a OneCycle [36] learning rate scheduler, which linearly decays from its peak value. Following the strategy in RVT [11], we employ a mixed batch training technique, where standard Backpropagation Through Time (BPTT) is applied to half of the batch samples, while Truncated BPTT (TBPTT) is applied to the other half. Data augmentation includes random horizontal flipping, zoom-in, and zoom-out operations. The event representation is constructed as a 5-channel voxel grid [45] based on a 50 ms time window. For the detection head, we utilize a Feature Pyramid Network (FPN) [24] for multi-scale feature fusion, along with the detection head from YOLOv6 [21], which incorporates distribution focal loss, classification loss, and regression loss.

To compare against prior works on the Gen1 dataset, we train our models with a batch size of 6, sequence length of 21, learning rate of $2e-4$ for $400k$ iterations. The training takes approximately 4 days on a single RTX 3090 GPU. On the 1 Mpx dataset, we train with a batch size of 8, sequence length of 5, and learning rate of $3e-4$ for $800k$ iterations on a single RTX 3090 GPU. On the eTram dataset, the model is trained for $400k$ iterations with a batch size of 8, a sequence length of 5, and an initial learning rate of $3e-4$.

Benchmark Comparisons. We compare the proposed neural network architecture with previous works on the Gen1 [5] and 1 Mpx [31] datasets, with the results summarized in Table 1. From Table 1, it can be concluded that the use of recurrent layers generally leads to better performance. S5-ViT-B [50] achieves competitive results by replacing recurrent layers with state-space models (SSM) [12, 35]. ERGO-12 [49] adaptively encodes spatiotemporal information from events, achieving

Table 1: Comparison with state-of-the-art methods on Gen1 and 1 Mpx datasets.

Method	Params	Backbone	Gen1 mAP	Time (ms)	1Mpx mAP	Time (ms)
Asynet [26]	11.4	Sparse CNN	14.5	-	-	-
AEGNN [32]	20.0	GNN	16.3	-	-	-
Spiking DenseNet [4]	8.2	SNN	18.9	-	-	-
Inception + SSD [16]	$> 60^*$	CNN	30.1	19.4	34.0	45.2
RRC-Events [3]	$> 100^*$	CNN	30.7	21.5	34.3	46.4
MatrixLSTM [2]	61.5	CNN + RNN	31.0	-	-	-
YOLOv3 Events [18]	$> 60^*$	CNN	31.2	22.3	31.6	49.4
RED [31]	24.1	CNN + RNN	40.0	16.7	43.0	39.3
ASTMNet [22]	$> 100^*$	CNN + RNN	46.7	35.6	48.3	72.3
ERGO-12 [49]	59.6	Transformer	<u>50.4</u>	69.9	46.0	100.0
RVT-B [11]	18.5	Transformer + RNN	47.2	10.2	47.4	<u>11.9</u>
Swin-T v2 [25]	21.1	Transformer + RNN	45.5	26.6	45.5	34.8
Nested-T [43]	22.2	Transformer + RNN	46.3	20.6	46.0	33.5
GET-T [30]	21.9	Transformer + RNN	47.9	16.8	48.4	18.2
SAST-CB [29]	18.9	Transformer + RNN	48.2	22.7	<u>48.7</u>	23.6
S5-ViT-B [50]	18.2	Transformer + SSM	47.7	**8.16**	47.8	**9.57**
Ours	26.4	CNN + RNN	**52.7**	<u>8.80</u>	**49.1**	13.3

Table 2: Comparison with state-of-the-art methods on the traffic monitoring dataset eTram.

Method	Backbone	mAP	Time (ms)
RVT-B [11]	Transformer+RNN	29.5	**10.88**
SAST-CB [29]	Transformer+RNN	<u>30.0</u>	23.07
S5-ViT-B [50]	Transformer+SSM	29.3	14.84
Ours	CNN+RNN	**33.0**	<u>13.05</u>

Table 3: Performance of different temporal information reuse methods on the Gen1 test set.

Fuse-Method	Feature-used	mAP	AP_{50}	Para. (M)
Base	-	50.7	80.6	22.4
Directly fusion	hidden	51.7	80.8	26.6
	cell	52.0	81.5	
DDConv	hidden	52.2	81.5	26.4
	cell	52.7	81.5	
DDConv + SE	hidden	52.3	81.7	26.4
	cell	52.7	81.7	

state-of-the-art performance without employing recurrent layers. It is worth noting that, unlike our CNN-RNN based approach, MatrixLSTM [2] applies LSTM units directly at the input level, while RED [31] and ASTMNet [22] utilize recurrent layers only in deeper network stages.

Our model achieves state-of-the-art performance with an mAP of 52.7 on the Gen1 dataset, and an mAP of 49.1 on the 1 Mpx dataset. Compared to other CNN-RNN methods in the table, our model achieves significantly higher performance while maintaining comparable or lower parameter counts. On the 1 Mpx dataset, it outperforms the second-best ASTMNet [22] by 0.8 mAP, and on the Gen1 dataset, it surpasses it by 6.0 mAP. Notably, our model is trained from scratch without requiring any pre-trained weights.

To further validate the generalizability of our model across datasets, we compared it on the sparser eTram dataset [37] against several Transformer- and SSM-based methods (RVT[11], SAST [29], S5-ViT [50]) that have demonstrated strong performance on the Gen1 [5] and 1 Mpx [31] datasets. As shown in Table 2, our method attains the highest detection accuracy on the eTram dataset [37] while maintaining a high inference speed, achieving an mAP that is 3.0 higher than that of the second-best SAST-CB [29].

Ablation Study. (1) Effectiveness of Decoupled Deformable-enhanced Recurrent Layer. Table 3 systematically investigates the effectiveness of our proposed Decoupled Deformable-enhanced Recurrent Block (DDRB). The baseline model ('Base') employs conventional ConvLSTM layers [33] within our custom backbone architecture without temporal enhancement mechanisms, achieving 50.7 mAP on the Gen1 dataset. This baseline performance inherently validates the fundamental efficacy of our backbone design. Furthermore, we attempt to enhance the base model by inserting a

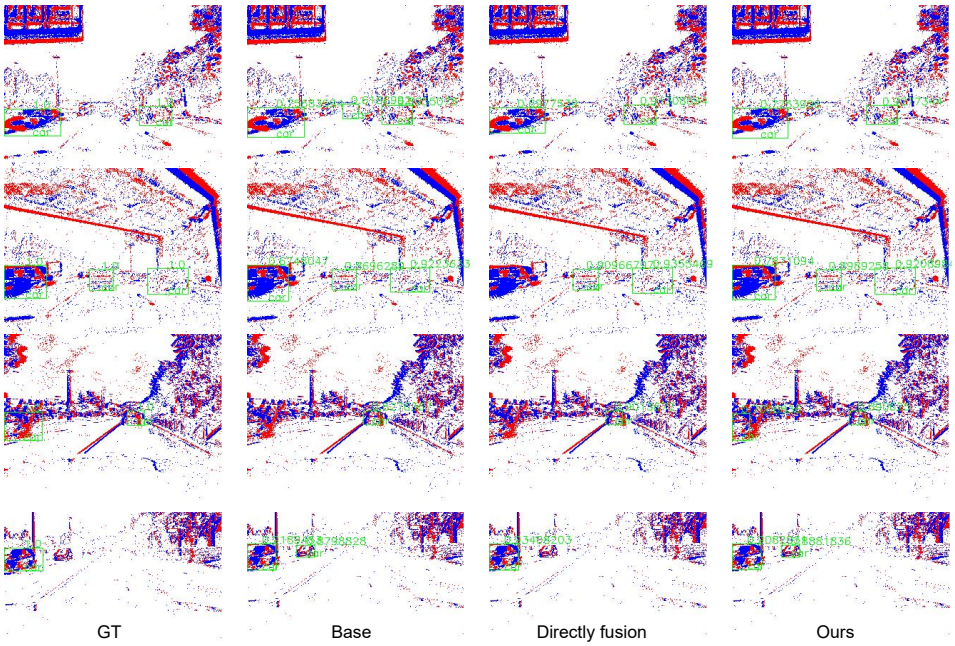


Figure 4: Visualizations of the Decoupled Deformable-enhanced Recurrent Layer (DDRL) ablation study on the Gen1 dataset.

5×5 convolution before each ConvLSTM to fuse the current event features with previous temporal information (called Directly Fusion in Table 3). The results show that using the hidden state leads to a 1.0 mAP improvement, while using the cell state yields a 1.3 mAP increase, indicating that further exploiting prior temporal information can indeed enhance detection accuracy.

Inspired by the divide-and-conquer principle, one of our principal innovations replaces these conventional convolutions with event-based decoupled deformable convolutions (called DDConv in Table 3). This modification achieves superior performance ($+0.5$ mAP with hidden states, $+0.7$ mAP with cell states) while reducing parameter count, establishing an optimal balance between model efficiency and detection accuracy. In addition, we validate the effectiveness of integrating a lightweight SE block [14], which provides further performance improvement without increasing model complexity.

(2) Effectiveness of the Proposed Backbone. To further investigate the effectiveness of the proposed backbone, we conducted experiments focusing on its key characteristics (Temporal Modeling at Lower Scales and Scale-Specific Spatiotemporal Encoding). All backbone variants in this study utilized 5-channel event voxel [45] as input and shared a YOLOv6 [21] detection head configuration. We maintained consistent feature downsampling ratios ($8\times$, $16\times$, $32\times$) and corresponding channel dimensions (128, 256, 512) at the detection head input across all configurations. Except for the RVT [11] backbone, all implementations employed cell state for temporal feature enhancement. The experimental results are systematically presented in Table 4.

Our initial investigation explored positioning the Decoupled Deformable-enhanced Recurrent Block (DDRB) in higher-dimensional spaces (as illustrated in Figure 6 (a)). Due to the increased number of channels at higher dimensions, the model parameters surged to 53.8M (an increase of 27.4M). However, this led to a 3.2 mAP drop, indicating that our low-dimensional spatiotemporal modeling strategy fundamentally aligns with event data characteristics. Additionally, we designed a single-branch feature extraction backbone based on our current backbone (with the placement of the recurrent layers unchanged, as shown in Figure 6 (b)). Although this slightly reduced the parameter count, it also led to a 1.4 drop in mAP. Furthermore, we compared our backbone with the RVT [11] backbone one of the representative classical Transformer-RNN methods. Without significantly increasing model complexity, our designed backbone achieved improvements of 4.7 in mAP and 4.2 in AP_{50}.

Visualizations. Figures 4 and 5 present partial visualization results from the two ablation studies. The comparative analysis demonstrates that our methodology, which strategically leverages temporal information through a divide-and-conquer principle capitalizing on event data characteristics, enhances model robustness for object detection across varying motion velocities. Furthermore,

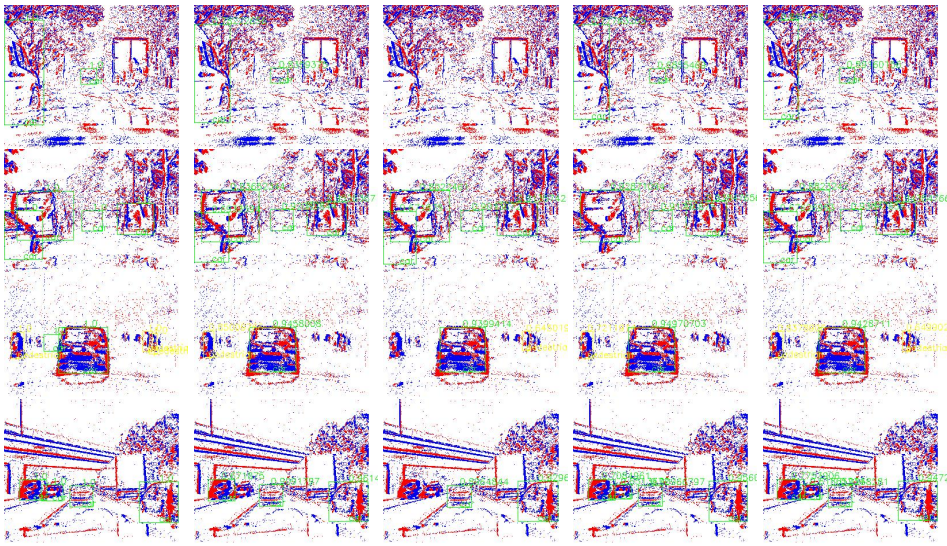


Figure 5: Visualization of the backbone ablation study on the Gen1 dataset.

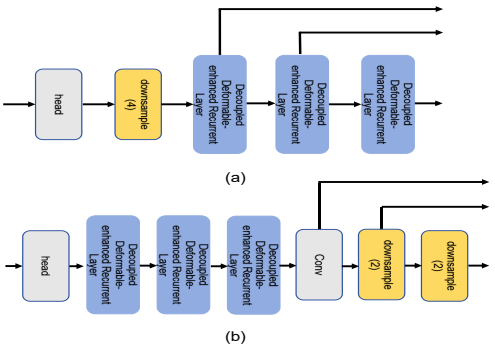


Figure 6: Backbone variants used in the ablation studies. (a) High-level Temporal Modeling. (b) Single-branch Encoding. The numbers after "downsample" indicate the downsampling ratio.

Table 4: Performance of different backbone types on the Gen1 test set.

Backbone Types	**mAP**	**AP$_{50}$**	**Para. (M)**
High-level Temporal Modeling	49.5	79.5	53.8
Single-branch Encoding	51.3	80.9	23.5
RVT [11] backbone	48.0	77.5	23.2
Ours	**52.7**	**81.7**	26.4

the incorporation of fine-grained temporal propagation mechanisms enables better performance in challenging scenarios involving partial occlusions or small object detection.

5 Conclusion

This paper revisits the architectural design of event-based object detectors by emphasizing precise temporal modeling over increased complexity. We demonstrate that placing recurrent modules at lower spatial scales enables effective capture of dense temporal patterns present in raw event streams. To further enhance motion alignment and feature quality, we introduce the Decoupled Deformable-enhanced Recurrent Layer (DDRL), which decouples motion estimation from feature fusion and leverages deformable convolution to adaptively align motion while suppressing task-irrelevant noise in low-dimensional features. Combined with scale-specific downsampling and feature fusion, our CNN–RNN architecture achieves competitive or superior performance to transformer-based models without relying on global attention mechanisms. These results highlight that enhancing temporal modeling at the feature extraction stage is key to advancing event-based vision and point toward designing temporally-aware yet structurally simple backbones for sparse, asynchronous data.

Acknowledgment

This work is partially supported by grants from the National Natural Science Foundation of China under contract No. 62302041.

References

[1] Julian Büchel, Gregor Lenz, Yalun Hu, Sadique Sheik, and Martino Sorbaro. Adversarial attacks on spiking convolutional neural networks for event-based vision. *Frontiers in Neuroscience*, 16:1068193, 2022.

[2] Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. Asynchronous convolutional networks for object detection in neuromorphic cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[3] Nicholas F. Y. Chen. Pseudo-labels for supervised learning on dynamic vision sensor data, applied to object detection under ego-motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[4] Loïc Cordone, Benoît Miramond, and Philippe Thierion. Object detection with spiking neural networks on automotive event data. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.

[5] Pierre De Tournemire, Davide Nitti, Etienne Perot, Davide Migliore, and Amos Sironi. A large scale event-based detection dataset for automotive. *arXiv preprint arXiv:2001.08499*, 2020.

[6] Ziluo Ding, Rui Zhao, Jiyuan Zhang, Tianxiao Gao, Ruiqin Xiong, Zhaofei Yu, and Tiejun Huang. Spatio-temporal recurrent networks for event-based optical flow estimation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 525–533, 2022.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022.

[9] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. Asynchronous, photometric feature tracking using events and frames. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 750–765, 2018.

[10] Daniel Gehrig and Davide Scaramuzza. Pushing the limits of asynchronous graph-based object detection with event cameras. *arXiv preprint arXiv:2211.12324*, 2022.

[11] Mathias Gehrig and Davide Scaramuzza. Recurrent vision transformers for object detection with event cameras. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13884–13893, 2023.

[12] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[14] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[15] Yuhuang Hu, Tobi Delbruck, and Shih-Chii Liu. Learning to exploit multiple vision modalities by using grafted networks. In *European Conference on Computer Vision*, pages 85–101. Springer, 2020.

[16] Massimiliano Iacono, Stefan Weber, Arren Glover, and Chiara Bartolozzi. Towards event-driven object detection with off-the-shelf deep learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9, 2018.

[17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *JMLR.org*, 2015.

[18] Zhuangyi Jiang, Pengfei Xia, Kai Huang, Walter Stechele, Guang Chen, Zhenshan Bing, and Alois Knoll. Mixed frame-/event-driven fast pedestrian detection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8332–8338, 2019.

[19] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Computer Science*, 2014.

[20] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E Shi, and Ryad B Benosman. Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1346–1359, 2016.

[21] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, et al. Yolov6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 2022.

[22] Jianing Li, Jia Li, Lin Zhu, Xijie Xiang, Tiejun Huang, and Yonghong Tian. Asynchronous spatio-temporal memory network for continuous event-based object detection. *IEEE Transactions on Image Processing*, 31:2975–2987, 2022.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014.

[24] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018.

[25] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 2022.

[26] Nico Messikommer, Daniel Gehrig, Antonio Loquercio, and Davide Scaramuzza. Event-based asynchronous sparse convolutional networks. In *European Conference on Computer Vision*, pages 415–431. Springer, 2020.

[27] Anton Mitrokhin, Cornelia Fermüller, Chethan Parameshwara, and Yiannis Aloimonos. Event-based moving object detection and tracking. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9, 2018.

[28] Diederik Paul Moeys, Federico Corradi, Emmett Kerr, Philip Vance, Gautham Das, Daniel Neil, Dermot Kerr, and Tobi Delbrück. Steering a predator robot using a mixed frame/event-driven convolutional neural network. In *2016 Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, pages 1–8, 2016.

[29] Yansong Peng, Hebei Li, Yueyi Zhang, Xiaoyan Sun, and Feng Wu. Scene adaptive sparse transformer for event-based object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16794–16804, 2024.

[30] Yansong Peng, Yueyi Zhang, Zhiwei Xiong, Xiaoyan Sun, and Feng Wu. Get: Group event transformer for event-based vision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6038–6048, 2023.

[31] Etienne Perot, Pierre De Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. Learning to detect objects with a 1 megapixel event camera. *Advances in Neural Information Processing Systems*, 33:16639–16652, 2020.

[32] Simon Schaefer, Daniel Gehrig, and Davide Scaramuzza. Aegnn: Asynchronous event-based graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12371–12381, 2022.

[33] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

[34] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1731–1740, 2018.

[35] Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022.

[36] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019.

[37] Aayush Atul Verma, Bharatesh Chakravarthi, Arpitsinh Vaghela, Hua Wei, and Yezhou Yang. etram: Event-based traffic monitoring dataset. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22637–22646, 2024.

[38] Dongsheng Wang, Xu Jia, Yang Zhang, Xinyu Zhang, Yaoyuan Wang, Ziyang Zhang, Dong Wang, and Huchuan Lu. Dual memory aggregation network for event-based object detection with learnable representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2492–2500, 2023.

[39] Xiao Wang, Chao Wang, Shiao Wang, Xixi Wang, Zhicheng Zhao, Lin Zhu, and Bo Jiang. Mambaevt: Event stream based visual object tracking using state space model. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[40] Nan Yang, Yang Wang, Zhanwen Liu, Meng Li, Yisheng An, and Xiangmo Zhao. Smamba: Sparse mamba for event-based object detection. *arXiv preprint arXiv:2501.11971*, 2025.

[41] Friedemann Zenke and Surya Ganguli. Superspike: Supervised learning in multilayer spiking neural networks. *Neural computation*, 30(6):1514–1541, 2018.

[42] Pengjie Zhang, Lin Zhu, Xiao Wang, Lizhi Wang, Wanxuan Lu, and Hua Huang. Ematch: A unified framework for event-based optical flow and stereo matching. *arXiv preprint arXiv:2407.21735*, 2024.

[43] Zizhao Zhang, Han Zhang, Long Zhao, Ting Chen, Sercan Ö Arik, and Tomas Pfister. Nested hierarchical transformer: Towards accurate, data-efficient and interpretable visual understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3417–3425, 2022.

[44] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. *arXiv preprint arXiv:1802.06898*, 2018.

[45] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 989–997, 2019.

[46] Lin Zhu, Xianzhang Chen, Lizhi Wang, Xiao Wang, Yonghong Tian, and Hua Huang. Continuous-time object segmentation using high temporal resolution event camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[47] Lin Zhu, Xiao Wang, Yi Chang, Jianing Li, Tiejun Huang, and Yonghong Tian. Event-based video reconstruction via potential-assisted spiking neural network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3594–3604, 2022.

[48] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9308–9316, 2019.

[49] Nikola Zubić, Daniel Gehrig, Mathias Gehrig, and Davide Scaramuzza. From chaos comes order: Ordering event representations for object recognition and detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12846–12856, 2023.

[50] Nikola Zubic, Mathias Gehrig, and Davide Scaramuzza. State space models for event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5819–5828, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, the abstract and introduction have clearly stated the contributions of our work.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, the limitation is discussed in our appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The code is included in the supplementary materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code is included in the supplementary materials.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The training and test details are included in the experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The experiment uses a fixed dataset and fixed seeds, and does not involve these.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, all details have been provided.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, we have checked it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes. Two datasets have been cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Yes, the training details of our model is presented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: NA

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Supplementary Experiments

This section primarily provides an additional analysis of the ablation experiments related to the backbone, followed by an investigation of the placement of the Decoupled Deformable-enhanced Recurrent Layer (DDRL) through experimental evaluation.

A.1 Supplementary Ablation Experiment

During ablation studies on the backbone architecture, we devised two specialized variants, *High-level Temporal Modeling* and *Single-branch Encoding*, explicitly tailored to investigate its critical characteristics: Temporal Modeling at Lower Scales and Scale-Specific Spatiotemporal Encoding. We also compared our backbone with the backbone of the classic Transformer-RNN method RVT [11]. However, we realized that high-dimensional temporal modeling allows for the parallel placement of the recurrent module (as shown in Figure 7 (a)). Therefore, under the same configuration, we conducted an additional experiment, High-level Temporal Modeling (Parallel), where Decoupled Deformable-enhanced Recurrent Block (obtained by removing the downsampling module from DDRL, as shown in Figure 7 (b)) is parallelly placed at high dimensions.

The experimental results are shown in Table 5. Whether the recurrent module is cascaded or parallelly placed at high dimensions, both configurations lead to an increase in parameters accompanied by a performance degradation. This may be because deep networks, through multiple downsampling and filtering operations, may overly smooth sparse event signals, weakening the recurrent module's ability to respond to critical temporal changes. Additionally, the effect of parallel placement of the recurrent module is less effective than cascading, possibly because in the parallel structure, each scale's recurrent module independently processes temporal information, lacking cross-scale temporal context transmission. The high-level semantic features are unable to leverage the fine-grained temporal variations in the low-level high-resolution features, resulting in fragmented temporal modeling. In our approach, recurrent layers are continuously positioned at lower-dimensional stages, aiming to preserve fine-grained temporal information and cross-scale sequential context. Simultaneously, a multi-branch feature extraction mechanism is introduced to flexibly retain salient information while mitigating the loss incurred during the downsampling process.

Table 5: Performance of different backbone types on the Gen1 [5] test set.

Backbone-Types	**mAP**	**AP$_{50}$**	**Para. (M)**
High-level Temporal Modeling	49.5	79.5	53.8
High-level Temporal Modeling (Parallel)	47.5	77.4	53.8
Single-branch Encoding	51.3	80.9	23.5
RVT [11] backbone	48.0	77.5	23.2
Ours	**52.7**	**81.7**	26.4

A.2 Analysis of the placement of the Decoupled Deformable-enhanced Recurrent Layer (DDRL)

To further explore the optimal placement of the Decoupled Deformable-enhanced Recurrent Layer, we attempted to place it consecutively at different scales (as shown in Figure 8). In the experiment, we still use 5-channel event voxels [45] as input, with FPN [24] performing multi-scale feature fusion and the YOLOv6 [21] detection head conducting detection. Additionally, we ensure that the feature downsampling factors for each backbone input to the detection head are 8, 16, and 32, corresponding to channel numbers of 128, 256, and 512, respectively. The models were trained on the Gen1 [5] dataset for 5 epochs with a batch size of 6 and learning rate of 2e-4.

The experimental results are shown in Table 6. *Low*, *Mid*, and *High* represent the placement of the Decoupled Deformable-enhanced Recurrent Layer at three different positions, from low-dimensional to high-dimensional. The numbers in parentheses indicate the downsampling factors corresponding to the features input to the three ConvLSTM [33]. For example, when placed at the lowest dimension, the downsampling factors for the features input to ConvLSTM are 2, 4, and 8, respectively. It can be observed that placing the Decoupled Deformable-enhanced Recurrent Layer at low dimensions

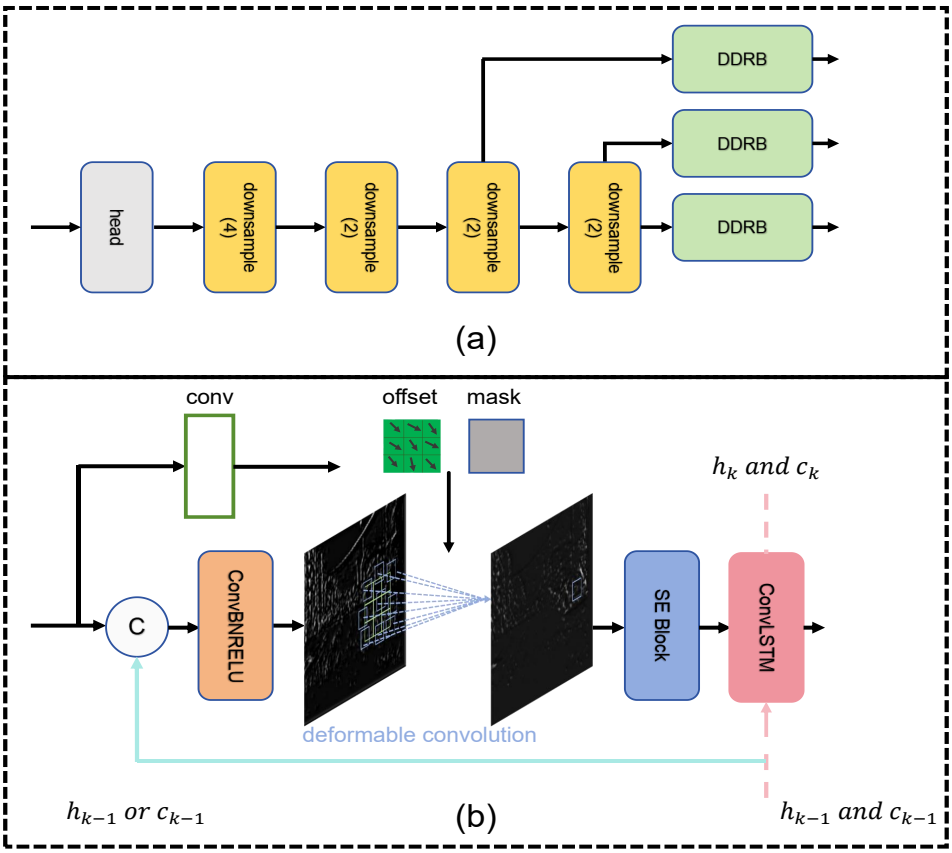


Figure 7: (a) High-level Temporal Modeling (Parallel). The number after 'downsample' indicates the downsampling ratio. (b) The architecture of Decoupled Deformable-enhanced Recurrent Block (DDRB).

yields better performance. This is likely because low-dimensional features typically preserve higher spatial resolution and finer-grained temporal information. Event camera data is inherently sparse and consists of asynchronous event streams. Shallow recurrent modules can directly model short-term motion patterns (such as edge movement and local brightness changes) on low-level abstract features, avoiding temporal blur in deep features caused by multiple downsampling operations.

Table 6: Performance difference of DDRL at different positions on the Gen1 [5] test set.

Placement	**mAP**	**AP$_{50}$**	**Para. (M)**
High (8, 16, 32)	49.5	79.5	53.8
Mid (4, 8, 16)	51.0	80.1	22.9
Low (2, 4, 8)	**52.0**	**81.5**	23.5

B Visualization

Visualization of temporal features. To more intuitively validate the effectiveness of our Decoupled Deformable-enhanced Recurrent Layer for temporal feature enhancement, we visualize the features output by the first and second recurrent layers (as shown in the Figure 9 and 10) and compare them with our base model (which uses ConvLSTM in the recurrent layers without enhancement). We did not choose to visualize the features from the third recurrent layer because, after a certain degree of downsampling, the features became too abstract. It can be observed that, since both models contain recurrent layers, the model is able to retrieve information from past events, even when objects

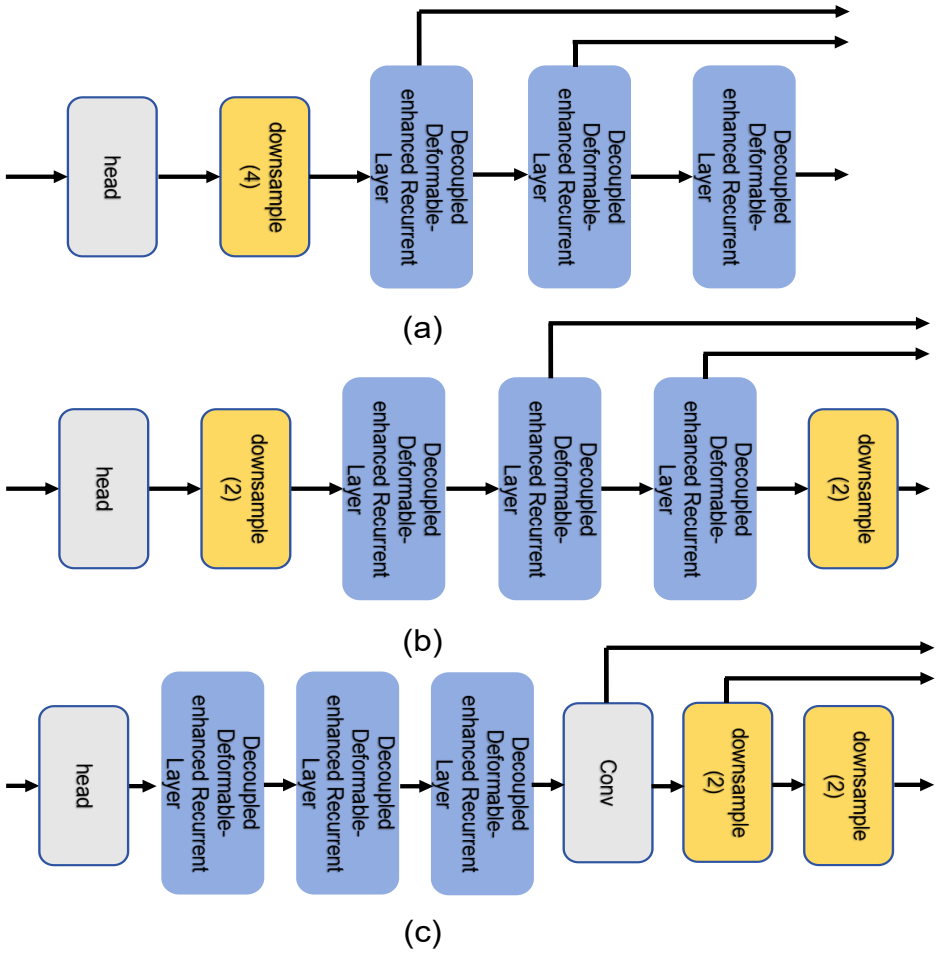


Figure 8: (a), (b), and (c) represent the placement of the Decoupled Deformable-enhanced Recurrent Layer from high-dimensional to low-dimensional, corresponding to *High*, *Mid*, and *Low*, respectively. The number after 'downsample' indicates the downsampling ratio. The 'Conv' used in (c) is specifically employed for channel dimension reduction.

gradually disappear in some scenarios. However, from the lower-dimensional features output by the first recurrent layer, it is evident that after enhancing the temporal features with our method, the details of the features are richer and the noise is significantly reduced. This may be the result of combining the motion information from events with deformable convolutions. From the higher-dimensional features output by the second recurrent layer, it is apparent that, compared to the base model, the enhanced model is able to focus more on the regions where objects are present.

As mentioned in the previous work RED [31], one of the important reasons for placing recurrent layers at high scales is to prevent recurrent layers from dynamically modeling low-level features that are unnecessary for the given task. However, the decoupled deformable-enhanced module we added before the recurrent layers can greatly reduce these unnecessary features through the smoothing effect of deformable convolution.

Visualization of deformable offsets. To further validate that the proposed Decoupled Deformable-enhanced Recurrent Layer (DDRL) effectively learns motion-aware spatial alignment, we visualize the learned deformable offsets (Δx_k) from the first DDRL using heat maps. The visualization is performed on the eTram dataset [37], a traffic monitoring benchmark where the event camera remains almost stationary during recording. This static setup allows us to more intuitively observe the motion patterns of moving vehicles and pedestrians without interference from ego-motion.

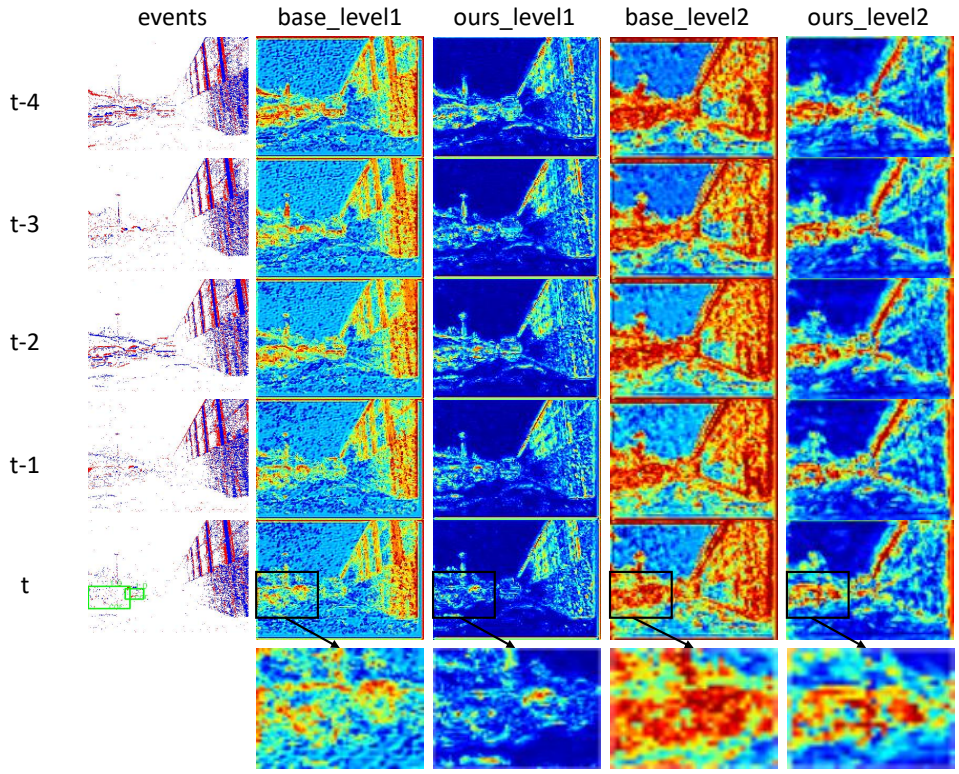


Figure 9: Visualization heatmap of the features output by the first (level1) and second (level2) recurrent layers on Gen1 [5]. 'base' is the variant of our model where temporal enhancement is not applied in the recurrent layers.

As illustrated in Figure 11, the learned offsets exhibit consistent and coherent motion directions aligned with object trajectories, confirming that the deformable module adaptively follows scene dynamics. Specifically, regions corresponding to moving vehicles show large, structured offsets oriented along the motion paths, while static background areas exhibit minimal displacement. This behavior demonstrates that the DDRL effectively decouples per-pixel motion estimation from feature fusion, adaptively aligning spatiotemporal features across consecutive event frames. These results empirically support our design motivation: the deformable convolution in DDRL captures fine-grained motion cues and spatially aligns event features according to actual object motion, thereby enhancing temporal consistency and detection accuracy in dynamic scenes.

C Limitations

Although the model we designed is capable of effectively leveraging the sparsity of events as well as the temporal and motion information contained within the events, resulting in promising performance, it also introduces a certain degree of computational complexity. Whether methods such as sparse convolutions can be employed to reduce the computational load without compromising performance is a question we are currently considering.

Moreover, we employ a relatively simple event representation that, although containing some temporal information, does not fully exploit the potential of event-based data. In future work, we aim to explore an event representation incorporating richer temporal information, thereby reducing the dependency of event based object detection on the parameter count and complexity of recurrent layers.

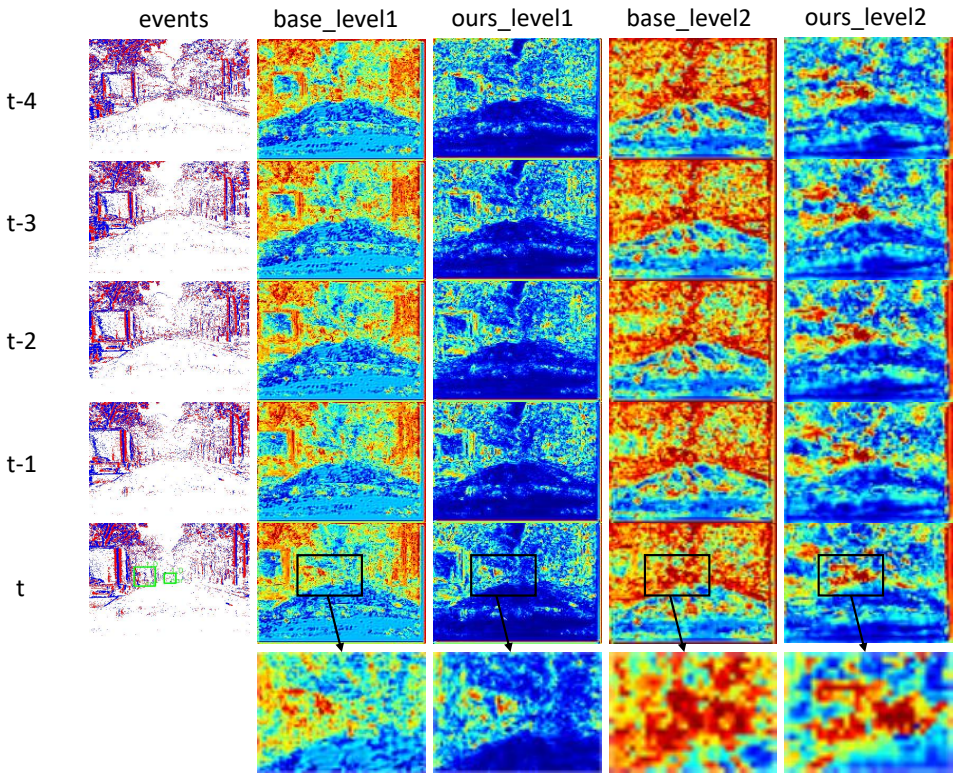


Figure 10: Visualization heatmap of the features output by the first (level1) and second (level2) recurrent layers on Gen1 [5]. 'base' is the variant of our model where temporal enhancement is not applied in the recurrent layers.

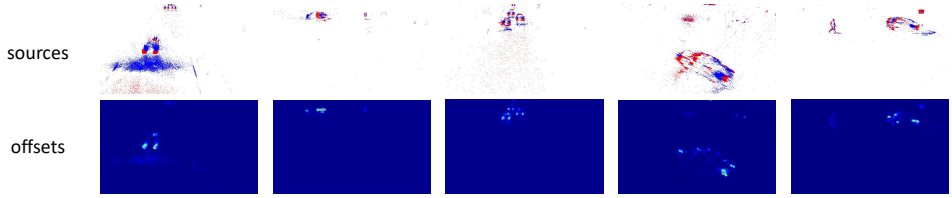


Figure 11: Visualization of the learned offsets from the first DDRL layer along with the corresponding source voxel visualization on eTram [37].