# Supplementary Materials
# EvRT-DETR: Latent Space Adaptation of Image Detectors for Event-based Vision

Dmitrii Torbunov[1], Yihui Ren[1], Animesh Ghose[1], Odera Dim[1], Yonggang Cui[1]

[1]Brookhaven National Laboratory, Upton, NY, USA

dtorbunov@bnl.gov, yren@bnl.gov, aghose@bnl.gov, dodera@bnl.gov, ycui@bnl.gov

## A. Training Setup

This section details the implementation of the I2EvDet framework as applied to event-based object detection. The training methodology reflects a two-stage adaptation approach: first establishing a robust spatial detector foundation then incorporating minimal temporal processing while preserving the pre-trained model's representation capabilities. This implementation demonstrates how mainstream detectors can be efficiently adapted to temporal data with minimal architectural changes

### A.1. RT-DETR

Our RT-DETR model is based on the reference PyTorch implementation[1] [13, 14]. We explore two RT-DETR configurations: RT-DETR-T (corresponds to RT-DETR-ResNet18) and RT-DETR-B (corresponds to RT-DETR-ResNet50). The RT-DETR configurations match the reference implementations, except two modifications: (1) we change the number of input channels of the backbones from 3 to 20, and (2) we do not use ImageNet pre-trained backbones. Table 1 summarizes the differences between the RT-DETR-ResNet18 and RT-DETR-ResNet50 configurations.

| Configuration | RT-DETR-T | RT-DETR-B |
|---|---|---|
| Backbone | ResNet-18 | ResNet-50 |
| FPN Features | $[128, 256, 512]$ | $[512, 1024, 2048]$ |
| Encoder Expansion | $\times 0.5$ | $\times 1.0$ |
| Decoder Layers | 3 | 6 |

Table 1. Comparison of the RT-DETR-T and RT-DETR-B Configurations.

We train all RT-DETR models for $400{,}000$ iterations using the Adam optimizer [7] with a batch size of 32 and a learning rate of $2 \times 10^{-4}$. Similar to the reference RT-DETR training, we maintain an exponential moving average (EMA) of the model weights with a momentum of $0.9999$. Unlike the reference RT-DETR training, we do not use any learning rate schedules or reduce the learning rate of the backbone relative to the encoder-decoder parts.

[1]commit `5b628eaa0a2fc25bdafec7e6148d5296b144af85`

**Environment.** The RT-DETR training is conducted with `PyTorch-2.2.2` and `torchvision-0.17.2` on a single NVIDIA RTX A6000 GPU.

### A.2. EvRT-DETR

The EvRT-DETR-T and EvRT-DETR-B models respectively extend RT-DETR-T and RT-DETR-B by adding ConvLSTM modules [11]. Our ConvLSTM blocks use a hidden dimension of 256 (matching the encoder feature maps) and a kernel size of 3. The temporal module outputs are integrated into the base model through residual connections with a scaling factor of 1.0 as we found learnable scaling parameters (similar to ReZero [1]) provide no measurable performance benefit on the Gen1 dataset.

The ConvLSTM modules are trained jointly for $200{,}000$ iterations with the Adam optimizer and a batch size of 8 while keeping the baseline RT-DETR models frozen. Each batch contains 8 short clips of consecutive frames: 21 frames for Gen1 dataset and 10 frames for 1Mpx dataset. The number of frames per clip follows RVT's approach [6], except we increase the number of frames for the 1Mpx dataset from 5 to 10 for better performance. Each batch contains 4 randomly sampled clips and 4 consecutive clips from videos.

Unlike the baseline RT-DETR training with a constant learning rate and EMA averaging, we do not apply EMA averaging to the ConvLSTM modules and use a learning rate scheduler instead. For simplicity, we rely on the RVT-inspired one-cycle LR scheduler [12]. Specifically, we use `PyTorch`'s `OneCycleLR` implementation of the scheduler, with the following parameters: maximum LR $2 \times 10^{-4}$, initial `div_factor` 20, final `div_factor` 500,

`pct_start` 0.0005, and annealing strategy "linear."

The ConvLSTM modules are trained in the same environment as the baseline RT-DETR networks using a single NVIDIA RTX A6000 GPU.

### A.3. YOLOX Baselines

We implement YOLOX baselines to demonstrate the generalizability of our I2EvDet framework beyond RT-DETR. We use the reference YOLOX backbone and head implementations[2] [4, 5]. The only architectural modification is changing the number of input channels of the backbone from 3 to 20.

To ensure fair comparison across architectures, we train YOLOX baselines using exactly the same setup as RT-DETR (cf. subsection A.1). Specifically, we train for 400,000 iterations using the Adam optimizer with a batch size of 32 and a learning rate of $2 \times 10^{-4}$.

The inference is performed with NMS threshold of $0.65$ and confidence threshold of $0.01$.

### A.4. I2EvDet on YOLOX

Following our I2EvDet framework, we extend the YOLOX baseline with temporal processing capabilities. We freeze the pre-trained YOLOX parameters and insert ConvLSTM modules between the PAFPN [8] neck and detection head.

The ConvLSTM modules follow the same configuration as EvRT-DETR: hidden dimension of 256, kernel size of 3, and residual integration with scaling factor 1.0. Training follows the identical protocol as EvRT-DETR with 200,000 iterations using the Adam optimizer and OneCycleLR scheduler (max LR $2 \times 10^{-4}$). Each batch contains 4 random and 4 consecutive clips of 21 frames (Gen1) or 10 frames (1Mpx).

### A.5. Notes on Two-Stage versus End-to-End Training

While developing the I2EvDet framework, we experimented with both end-to-end and two-stage training approaches. For the end-to-end training, we combined the RT-DETR model with the ConvLSTM temporal adapters and trained them jointly. Our experiments indicate that the two-stage training approach enables significantly faster convergence compared to end-to-end training with much higher stability. Moreover, in our experiments, the two-stage training has no performance downsides compared to the end-to-end training.

Based on these observations, we adopt the two-stage training approach in this work. An additional benefit of two-stage training is it enables experimentation with different temporal modules on the same spatial detector baseline, accelerating development and providing clear separation between spatial and temporal performance components.

[2]commit `ac58e0a5e68e57454b7b9ac822aced493b553c53`

We note that our end-to-end training experiments did not involve extensive hyperparameter exploration, so it is possible that alternative hyperparameter configurations may improve end-to-end training performance. However, the two-stage approach has proven more robust and practical for our framework development.

### A.6. Data Augmentation Strategy

While data augmentation is standard practice in computer vision [2, 3], its application to event-based data has been limited [10]. Our adaptation framework treats EBC data as image-like frames, allowing us to leverage established augmentation techniques from mainstream computer vision.

Our augmentation strategy employs a standard chain of geometric transformations and random erasing as summarized in Table 2. This approach bridges conventional image augmentation practices with the unique characteristics of event data, supporting the broader goal of adapting mainstream techniques to event-based vision.

| Augmentation | Magnitude | Probability |
|---|---|---|
| Random Horizontal Flip | - | 0.5 |
| Random Rotation | $\pm 30°$ | 0.6 |
| Random Translation | $\pm 0.5$ | 0.6 |
| Random Scale | $(0.5, 1.5)$ | 0.6 |
| Random Shear | $\pm 30°$ | 0.6 |
| Random Erasure | - | 0.4 |

Table 2. Standard augmentation chain adapted for event-based object detection. Each transformation is applied sequentially with the indicated probability to bridge mainstream vision techniques with event data processing.

Table 2 shows our augmentation strategy, which leverages standard transformations from the `torchvision` package (`0.17.2`) [9]. Each augmentation is applied independently with its corresponding probability. When applied, its magnitude is randomly sampled from the specified range. For Random Erasure, we preserve the object labels while erasing image regions, maintaining detection supervision even in partially occluded scenarios.

**Temporal Consistency in Augmentations.** Our two-stage adaptation approach requires different augmentation strategies for each stage. For the base detector (RT-DETR) training on individual frames, we apply augmentations independently to each frame following standard practice.

For the temporal adaptation stage, we carefully preserve temporal consistency by applying identical geometric transformations (flip, rotation, scale, translation, shear) across all frames in a video clip while still allowing per-frame variations through random erasing. This ensures that the temporal module learns meaningful motion patterns rather than

artificially induced movements from inconsistent augmentations.

## B. Supplementary Ablation Studies

This section provides additional ablation studies examining data augmentation strategies for event-based object detection and spatial context size in our temporal adaptation modules.

### B.1. Impact of Augmentation on Base Detector Performance

When applying mainstream object detectors to event data, appropriate data augmentation becomes crucial for effective domain transfer. Our experiments show that without a proper augmentation strategy, the base detector performance is significantly compromised.

| Model | mAP (%) | $mAP_{50}$ (%) | $mAP_{75}$ (%) |
|---|---|---|---|
| RT-DETR-B (ours) | **47.6** | **76.3** | 49.5 |
| RT-DETR-B (no augs) | 38.6 | 62.8 | 39.8 |
| RT-DETR-B (-Rotation) | 46.8 | 74.4 | 48.8 |
| RT-DETR-B (-Scale) | 45.6 | 73.8 | 47.2 |
| RT-DETR-B (-Translation) | 45.0 | 72.6 | 46.2 |
| RT-DETR-B (-Shear) | **47.6** | 75.9 | **49.7** |
| RT-DETR-B (-Erase) | 46.8 | 75.1 | 48.7 |

Table 3. Impact of data augmentation techniques on base detector performance for the Gen1 dataset.

Table 3 quantifies the contribution of different augmentation techniques to our adaptation framework. Without augmentations, RT-DETR-B performance drops by 9 mAP points, highlighting their critical role in successful adaptation. Among individual transformations, spatial manipulations (translation and rescaling) provide the largest gains, suggesting that scale and position invariance are particularly important when adapting image detectors to event data. Random rotations and erasure techniques offer moderate improvements, while shear transformations show minimal impact.

## C. Spatial Context Size in Temporal Processing

| Model | mAP (%) | $mAP_{50}$ (%) | $mAP_{75}$ (%) |
|---|---|---|---|
| EvRT-DETR-B (KS=1) | 52.3 | 81.4 | 55.4 |
| EvRT-DETR-B (KS=3) | **52.7** | **82.0** | **56.0** |
| EvRT-DETR-B (KS=5) | 52.0 | 81.3 | 54.8 |

Table 4. Effect of ConvLSTM Kernel Size. Ablation study on Gen1 dataset showing optimal temporal adaptation performance with $3 \times 3$ kernels.

Previous work on event-based object detection with ConvLSTM models, specifically RVT [6], found optimal performance with $1 \times 1$ ConvLSTM kernels (effectively pointwise LSTMs). However, our adaptation approach shows different optimal characteristics. As shown in Table 4, EvRT-DETR achieves best performance with $3 \times 3$ kernels, suggesting that spatial context is valuable when adapting frozen RT-DETR features. This demonstrates how adaptation design choices may differ from specialized architectures built from the ground up. Increasing to $5 \times 5$ kernels degrades performance, suggesting that the additional parameters and receptive field expansion do not provide beneficial information for the task given the available training data.

## D. Detailed YOLOX Results

To demonstrate that our findings extend beyond transformer-based architectures, we evaluate our I2EvDet framework on YOLOX [4], a CNN-based detector that has been used in other EBC applications [6, 15].

| Model | mAP (%) | $mAP_{50}$ (%) | $mAP_{75}$ (%) |
|---|---|---|---|
| RT-DETR-B | **47.6** | **76.3** | **49.5** |
| YOLOX-T | 36.0 | 59.9 | 36.3 |
| YOLOX-S | 37.0 | 61.1 | 37.6 |
| YOLOX-L | 43.1 | 69.6 | 44.7 |
| YOLOX-X | 43.4 | 69.7 | 44.7 |
| EvRT-DETR-B | **52.7** | **82.0** | **56.0** |
| EvYOLOX-T | 42.4 | 71.9 | 42.8 |
| EvYOLOX-S | 43.6 | 73.1 | 44.4 |
| EvYOLOX-L | 46.6 | 75.4 | 48.2 |
| EvYOLOX-X | 47.8 | 75.7 | 50.0 |

Table 5. Performance comparison of RT-DETR and YOLOX variants with and without our I2EvDet temporal adaptation on the Gen1 dataset. The consistent improvements across all architectures and model sizes demonstrate the generalizability of our adaptation framework beyond transformer-based detectors to CNN-based architectures.

We apply our two-stage training approach to multiple YOLOX variants, using identical training configurations to our RT-DETR experiments. Table 5 shows that YOLOX models achieve respectable baseline performance on the Gen1 dataset with YOLOX-X reaching 43.4 mAP, which remains below RT-DETR-B's 47.6 mAP.

The I2EvDet framework provides substantial improvements across all YOLOX variants with gains ranging from 4.4 to 6.4 mAP. These consistent improvements demonstrate that temporal adaptation benefits extend across different architectural paradigms, validating our framework's generalizability. While transformer-based RT-DETR achieves higher absolute performance than CNN-based YOLOX variants, both architecture families benefit significantly from our temporal adaptation approach.

These results confirm that the I2EvDet framework represents a general strategy for adapting image-based detectors

to temporal domains with broad applicability beyond transformer architectures. Testing with the most recent YOLO iterations could be an interesting direction for future work but is beyond the scope of this current study.

## E. Adaptation Considerations for Higher Resolution Event Data (1Mpx)

This section examines specific adaptation factors for the high-resolution 1Mpx dataset, focusing on resolution reduction techniques and temporal context length.

### E.1. Resolution Adaptation Strategy

When adapting mainstream detectors to high-resolution event data, appropriate downsampling techniques become critical. For consistency with prior work, we reduce 1Mpx frames from $(720, 1280)$ to $(360, 640)$, but our investigation reveals that the interpolation method significantly impacts adaptation performance.

| Model | mAP (%) | $mAP_{50}$ (%) | $mAP_{75}$ (%) |
|---|---|---|---|
| RT-DETR-B (nearest) | 42.3 | 71.8 | 42.3 |
| RT-DETR-B (bilinear) | **45.2** | **75.1** | **46.0** |
| RT-DETR-B (bicubic) | 43.1 | 72.2 | 43.3 |

Table 6. Impact of interpolation methods on base detector performance for the 1Mpx dataset downsampling.

Table 6 depicts how the nearest-neighbor interpolation substantially degrades RT-DETR performance, while bilinear interpolation yields optimal results. Interestingly, despite its theoretical advantages for natural images, bicubic interpolation proves less effective for event data. These findings align with observations from AEC [10] and highlight the importance of selecting appropriate domain transfer techniques when adapting mainstream vision models to event data.

### E.2. Temporal Context Length for Effective Adaptation

The temporal dimension represents a critical aspect of our adaptation framework. While prior work like RVT [6] uses 5-frame clips for temporal training, our experiments indicate that expanded temporal context benefits the adaptation process.

| Model | mAP (%) | $mAP_{50}$ (%) | $mAP_{75}$ (%) |
|---|---|---|---|
| EvRT-DETR (5 frames) | 49.8 | 80.7 | 51.8 |
| EvRT-DETR (10 frames) | **50.1** | **80.9** | **52.1** |

Table 7. Effect of temporal clip length on adaptation performance for the 1Mpx dataset.

Table 7 demonstrates the improvement achieved by extending the temporal window to 10 frames during the adaptation phase. This finding implies that providing longer temporal context during training allows the model to better capture persistent object representations across the temporal dimension, which is particularly important for event data where objects may be incompletely represented in shorter time windows.

## F. Comprehensive Evaluation Metrics

To provide complete experimental validation, this section presents detailed performance metrics and additional visualizations of our models across both Gen1 and 1Mpx datasets.

### F.1. Detailed COCO Metrics

Table 8 provides comprehensive COCO evaluation metrics including mean Average Precision (mAP) and mean Average Recall (mAR) at different Intersection over Union (IoU) thresholds. I2EvDet's temporal adaptation consistently improves performance across all metrics and datasets, demonstrating the robustness of our approach.

### F.2. Additional Visualizations

Figure 1 presents qualitative detection results on diverse 1Mpx automotive scenarios across varied lighting conditions and object configurations. Both RT-DETR and EvRT-DETR demonstrate effective adaptation to event-based data representations, with comparable performance on dynamic scenes where objects generate sufficient event data through motion.

Figure 2 illustrates the critical advantage of temporal memory during motion transitions. As vehicles stop at an intersection and event generation becomes sparse, frame-based detection degrades significantly while our temporal adaptation maintains consistent object localization by leveraging historical information. This sequence exemplifies the fundamental challenge that motivates our I2EvDet framework and demonstrates its effectiveness in real-world automotive scenarios.

## References


[1] Thomas Bachlechner, Bodhisattwa Prasad Majumder, Huanru Henry Mao, Gary Cottrell, and Julian J. McAuley. ReZero is all you need: fast convergence at large depth. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021*, pages 1352–1361. AUAI Press, 2021. 1

[2] Ekin Dogus Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. AutoAugment: learning augmentation policies from data. *CoRR*, abs/1805.09501, 2018. 2

[3] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*, pages 3008–3017. Computer Vision Foundation / IEEE, 2020. 2

| Model | Data | mAP (%) | $mAP_{50}$ (%) | $mAP_{75}$ (%) | mAR (%) | $mAR_{50}$ (%) | $mAR_{75}$ (%) |
|---|---|---|---|---|---|---|---|
| RT-DETR-T (ours) | Gen1 | 46.0 | 74.4 | 47.2 | 38.3 | 53.2 | 38.3 |
| RT-DETR-B (ours) | Gen1 | 47.5 | 76.3 | 49.5 | 39.9 | 54.6 | 42.1 |
| EvRT-DETR-T (ours) | Gen1 | 52.3 | 81.4 | 55.2 | 44.2 | 59.2 | 51.8 |
| EvRT-DETR-B (ours) | Gen1 | **52.7** | **82.0** | **56.0** | **45.1** | **59.3** | **53.3** |
| RT-DETR-T (ours) | 1Mpx | 44.1 | 74.2 | 44.3 | 33.5 | 50.7 | 52.6 |
| RT-DETR-B (ours) | 1Mpx | 45.2 | 75.1 | 46.0 | 35.1 | 51.8 | 53.5 |
| EvRT-DETR-T (ours) | 1Mpx | 49.9 | **81.0** | 51.6 | 38.6 | 55.5 | **62.4** |
| EvRT-DETR-B (ours) | 1Mpx | **50.1** | 80.9 | **52.1** | **39.0** | **55.6** | 61.7 |

Table 8. **Complete COCO Evaluation Metrics.** Comprehensive performance comparison showing mAP and mAR metrics at IoU thresholds of 0.5:0.95 (default), 0.5, and 0.75 for all model variants on Gen1 and 1Mpx datasets. Our temporal adaptation consistently improves both precision and recall across all thresholds and datasets, validating the robustness of the I2EvDet framework.

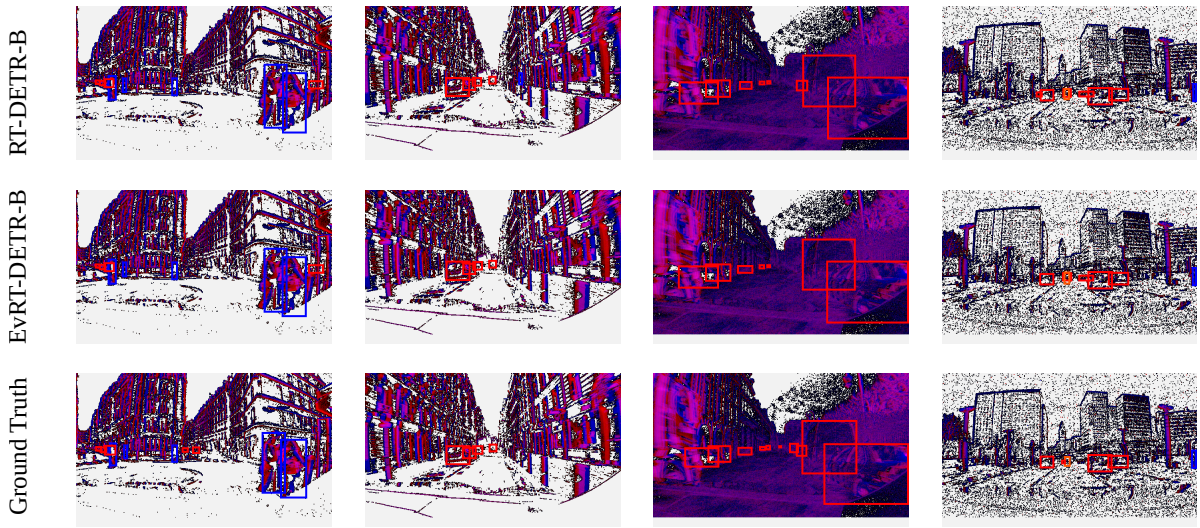


Figure 1. **Qualitative detection results on diverse 1Mpx automotive scenarios.** Top row: RT-DETR predictions. Middle row: EvRT-DETR predictions. Bottom row: Ground truth annotations. Bounding box colors indicate object classes: red (cars), blue (pedestrians), orange (two-wheelers). Both methods perform well on dynamic scenes with moving objects, demonstrating the effectiveness of our base RT-DETR adaptation to event-based data across varied lighting conditions and object configurations


[4] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: exceeding YOLO series in 2021. *CoRR*, abs/2107.08430, 2021. 2, 3

[5] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX is a high-performance anchor-free YOLO, exceeding yolov3 v5 with MegEngine, ONNX, TensorRT, ncnn, and OpenVINO supported. https://github.com/Megvii-BaseDetection/YOLOX, 2024. 2

[6] Mathias Gehrig and Davide Scaramuzza. Recurrent vision transformers for object detection with event cameras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 13884–13893. IEEE, 2023. 1, 3, 4

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 1

[8] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8759–8768. Computer Vision Foundation / IEEE Computer Society, 2018. 2

[9] TorchVision maintainers and contributors. TorchVision: PyTorch's computer vision library. https://github.com/pytorch/vision, 2016. 2

[10] Yansong Peng, Yueyi Zhang, Peilin Xiao, Xiaoyan Sun, and Feng Wu. Better and faster: Adaptive event conversion for event-based object detection. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 2056–2064. AAAI Press, 2023. 2, 4

[11] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung,

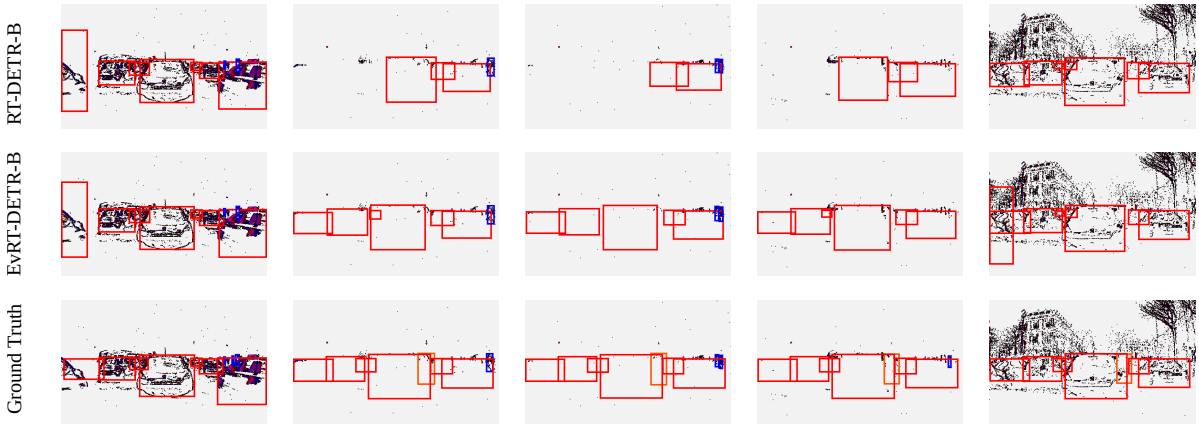


Figure 2. **Temporal sequence demonstrating event-based detection challenges during motion transitions**. A vehicle approaches an intersection, stops (creating sparse event data), then resumes motion. Frames shown every 100 frames for visualization clarity. Top row: RT-DETR predictions degrade significantly during stationary periods due to minimal event generation. Middle row: EvRT-DETR maintains more consistent detection by leveraging temporal memory from previous frames when objects were actively generating events. Bottom row: Ground truth annotations. Bounding box colors indicate object classes: red (cars), blue (pedestrians), orange (two-wheelers). While our temporal module substantially improves detection consistency during low-activity periods, challenges remain for heavily occluded objects, highlighting opportunities for future work

Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 802–810, 2015. 1

[12] 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*, pages 369–386. SPIE, 2019. 1

[13] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Official RT-DETR (RTDETR paddle pytorch), Real-Time DEtection TRansformer, DETRs beat YOLOs on Real-time object detection. https://github.com/lyuwenyu/RT-DETR. 1

[14] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. DETRs beat YOLOs on Real-time object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 16965–16974. IEEE, 2024. 1

[15] Nikola Zubic, Mathias Gehrig, and Davide Scaramuzza. State space models for event cameras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 5819–5828. IEEE, 2024. 3