Event-Aware Distilled DETR for Object Detection in an Automotive Context

Djessy Rossi[1], Pascal Vasseur[1], Fabio Morbidi[1], Cédric Demonceaux[2], François Rameau[3]

***Abstract*— Autonomous driving systems require robust object detection in complex environments. Event cameras outperform RGB cameras under challenging lighting conditions, but face limitations due to the scarcity of available datasets and lack of specialized training. To narrow the gap between RGB- and event-based detection accuracy and avoid the high complexity of real-time RGB-event fusion, in this paper, we propose a knowledge distillation framework. Our approach uses both modalities during training but relies solely on sparse event data at inference and transfers knowledge from a robust RGB-based teacher model. We build on the success of DETR (DEtection TRansformer) and we leverage an event-aware masked knowledge distillation mechanism, to boost event-based detection accuracy. Experiments on the DSEC-DET dataset demonstrate that our method not only excels in challenging driving scenarios where RGB images are unreliable, but also surpasses the state-of-the-art in event-based object detection.**

MULTIMEDIA MATERIAL

Our open-source code is available on GitHub at: github.com/djessy1998/EA-DETR.git
Qualitative results are available in the video: www.youtube.com/watch?v=MgFWLxA0IeE

I. INTRODUCTION

Event-based cameras are bio-inspired sensors, which record visual information in a fundamentally different way from conventional cameras. Instead of capturing frames at a fixed rate, event-based cameras output a continuous stream of events corresponding to pixel-level brightness changes. The pixel-level detection of luminance changes guarantees ultra-high temporal resolution, allowing fast-moving objects in dynamic scenes to be captured with minimal latency. In addition, the high dynamic range ensures consistent performance in variable/adverse lighting conditions. The emergence of compact, high-resolution event cameras from companies like Prophesee and iniVation has significantly impacted computer vision and robotic perception. In spite of their advantages, using an event camera to detect moving objects remains an open issue [1], [2], due to the unique nature of input data.

To tackle this challenge, deep learning models have been explored, but their performance varies widely depending on the modality. Convolutional Neural Networks (CNNs) [3] and Vision Transformers (ViTs) [4] are highly optimized for RGB data due to large-scale pre-training on high-resolution frame-based datasets. In contrast, Spiking Neural Networks (SNNs) [5]–[7] and Graph Neural Networks (GNNs) [8]–[10] are more suited for event data, as they can better handle the sparse and asynchronous nature of events. Nevertheless, the event-based models still underperform when compared to RGB-based ones, in terms of accuracy. In [11], [12] it has been shown that event cameras excel in automotive scenarios involving fast motion or sudden lighting changes, where RGB models struggle. However, the lack of large annotated event datasets and specialized deep learning architectures prevents event models from achieving comparable performances. Consequently, while event cameras hold great potential, significant work is needed to close the performance gap with established RGB-based architectures. This raises a critical question:

Can we take advantage of a high-precision RGB-based model to enhance the performance of an event-based model?

While it is possible to fuse RGB images with sparse event information and train a model with them [13]–[15], this solution suffers from a number of limitations. First, it requires two extrinsically-calibrated sensors to be installed on a vehicle, both at the training and inference stage. Second, the RGB and event cameras must be synchronized, which is, generally, a tedious and error-prone process. Finally, the volume of RGB-event data can become very large over time and difficult to handle, thus adversely impacting the performance of object-detection algorithm. However, instead of pursuing this path, we argue that it is possible to achieve performance comparable to or even better than that of RGB-only models by relying solely on event cameras at inference. In this paper, our goal is to address these issues and to narrow the accuracy gap between the RGB and event-based models, by utilizing the RGB information, *indirectly*. In fact, we leverage the knowledge of a model pre-trained on a large RGB dataset, during the training of a purely event-based object-detection network. Taking advantage of knowledge distillation for training event-based models, is justified by the large availability of annotated RGB datasets. By employing an *event-aware masked knowledge distillation* mechanism, we take advantage of the sparse nature of event data, but also of the robustness of consolidated RGB models. To maximize accuracy, we evaluated the impact of knowledge distillation on one of the most popular and powerful object-detection models: DETR (DEtection TRansformer) [16].

In summary, the original contribution of this paper is threefold:

1) We show how DETR (which is originally intended for RGB images) can be adapted to event data thanks to

[1]MIS laboratory, Université de Picardie Jules Verne, Amiens, France. Emails: {firstname.lastname}@u-picardie.fr
[2]ICB UMR 6303, Université Bourgogne Europe, CNRS, Dijon, France. Email: cedric.demonceaux@u-bourgogne.fr
[3]SUNY Korea, Incheon, South Korea. Email: francois.rameau@sunykorea.ac.kr

our cross-modal RGB-event distillation. We called this model *EA-DETR* (Event-Aware DETR),
2) EA-DETR is successfully validated on the DSEC-DET dataset, derived from DSEC [17], and on a new extract that we called Hard-DSEC-DET, which collects challenging scenarios for RGB cameras,
3) EA-DETR delivers a strong performance on both DSEC-DET and Hard-DSEC-DET, particularly excelling at detecting medium to large objects.

The remainder of this paper is organized as follows. In Sect. II, we discuss related work on RGB- and event-based object detection, we examine the existing knowledge distillation techniques, and provide an overview of DETR architecture. In Sect. III, we present our RGB-event distillation model, EA-DETR. In Sect. IV, EA-DETR is validated on the DSEC-DET dataset, considering different driving scenarios. Finally, in Sect. V, the main contributions of the paper are summarized and possible avenues for future research are discussed.

II. Related Work

A. RGB and event-based object detection

Object detection using event data is an open problem. Some researchers have developed neural networks tailored to event cameras to fully leverage their high temporal resolution and asynchronicity. Schaefer *et al.* [10] modeled events as graph points for object detection, while Cordone *et al.* [5] introduced a novel voxel cube encoding for SNNs to improved detection accuracy. Similarly, Gehrig & Scaramuzza [4] proposed Recurrent Vision Transformers (RVTs), which combine temporal aggregation, convolutional priors, self-attention, and LSTM (Long Short-Term Memory) cells to enhance event-based detection. Despite these advancements, event-based models generally lag behind RGB models in terms of accuracy.

To address this issue, other researchers have focused on RGB-event fusion. Zhou *et al.* [13] introduced RENet, a network designed for moving object detection in autonomous driving, which employs temporal multi-scale aggregation and bi-directional feature fusion to enhance accuracy. Tomy *et al.* [14] used a voxel grid representation for events and a dual feature extraction network to improve robustness under adverse conditions. Liu *et al.* [18] introduced SFNet for object detection in traffic conditions, which incorporates Speed Invariant Frames (SIF) and an Adaptive Feature Complement Module (AFCM) for effective cross-modal fusion under varying illumination. However, as these methods rely on RGB data during inference, they suffer from two main drawbacks. First, before the events being processed, one should wait for the acquisition of an RGB image, which introduces an inevitable delay. Second, the need for two sensors during the inference step, increases the overall complexity and cost of the system.

Another way to indirectly integrate RGB data for object detection without the previous limitations, is via *knowledge distillation*, as detailed in the next section.

B. Knowledge distillation for object detection

Knowledge distillation (KD) is a well-known technique which consists in transferring the "knowledge" from a larger, more complex model (Teacher) to a smaller, more efficient model (Student). The main goal is to achieve the same performances as the Teacher, while using reduced computational resources. Using RGB images, Chen *et al.* [19] introduced the first end-to-end trainable framework for compact multi-class object detection: it includes a distillation loss on the backbone, a classification head loss, and a regression head loss between the Teacher and the Student. This work also revealed that the imbalance between the number of background and foreground pixels is critical for knowledge distillation. To address this issue, several solutions have been proposed in the literature [20]–[22]. In [20], the authors used distillation with an L_2 loss only at locations sampled by a Region Proposal Network (RPN), while in [21], a mask is created via the model's attention, for attention-guided distillation. Finally, in [22], ground-truth bounding boxes are directly used to mask an L_2 loss.

Recently, motivated by the success of ViTs, the research community has turned its attention to specialized knowledge distillation techniques, for models in the DETR family [23]–[25]. Notably, Chang *et al.* [26] developed a method that utilizes DETR's queries and the Hungarian matching algorithm, as a means to distill knowledge.

Finally, knowledge distillation has been used with two (or more) complementary modalities, to enhance model accuracy. For example, Kruthiventi *et al.* [27] adapted the knowledge distillation mechanism in [19] to a thermal and an RGB camera, showing improved object-detection accuracy in low-light conditions.

Despite the progress made in this domain, many existing methods introduce significant architectural complexity. For example, object-centric methods [28], utilize multi-stage pipelines with coarse- and fine-level feature alignment, slot attention modules, and object relation distillation, which can be computationally intensive. Similarly, line-segment-based techniques [29] rely on scene-level affinity alignment and edge-specific feature masking, making them well-suited for line detection, but less adaptable to general object detection tasks. In contrast, our method employs a simpler and more scalable event-aware backbone distillation. By generating a Binary Events Mask (BEM) and applying a masked MSE loss, we reduce the impact of foreground-background imbalance without requiring specialized alignment modules. This allows us to effectively leverage the sparsity of event data, thus ensuring efficient training and inference for dynamic scenarios, such as urban traffic. Compared to multi-stage frameworks, our method strikes a balance between simplicity, computational efficiency, and accuracy.

C. DEtection TRansformer (DETR)

Transformers have been introduced by Vaswani *et al.* [30] for natural language processing. This prototypal model in [30], has been successively adapted to image classification [31] and object detection [16]. DETR streamlines the object-detection pipeline by eliminating

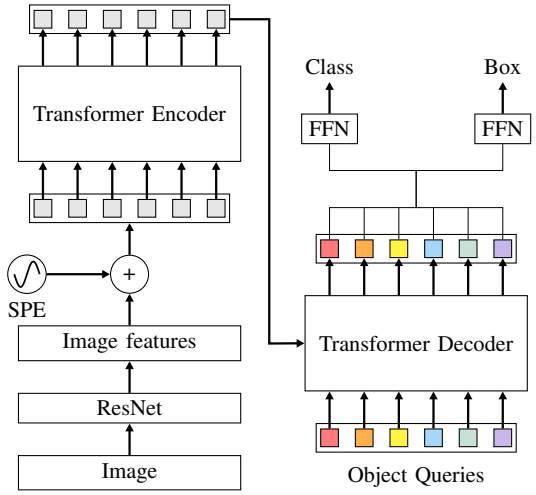

Fig. 1: Overview of the classical DETR architecture [16]. FFN and SPE stand for Feed-Forward Network and Spatial Positional Encoding, respectively.

the need for RPNs, which were typically used in earlier architectures, such as Faster R-CNN [32]. In contrast, DETR employs a global attention mechanism to process the entire image at once, and the object-detection problem is translated into a set prediction problem. Each element of this set represents a detected object, characterized by a bounding box and a class label (see Fig. 1 for an overview).

In this paper, we propose a cross-modal knowledge distillation method for RGB and event data, which is tailored to DETR. In particular, as detailed in the next section, we introduce a novel method for distilling knowledge from a Teacher to a Student model, by leveraging event information.

III. Proposed method: EA-DETR

This section begins by providing the necessary background to understand our method, followed by a brief description of the event representation used throughout the paper. The section concludes with a detailed explanation of our knowledge distillation mechanism. For later reference, Fig. 2 reports the overall training pipeline of EA-DETR.

A. Background

Since our method is based on DETR, it is worth briefly recalling how it processes information from input to output (see Fig. 1)

Given an input image $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{H\times W\times 3}$, DETR first uses a CNN backbone to extract feature maps $\mathbf{F} \in \mathbb{R}^{H_f\times W_f\times C}$, which are flattened into a sequence of vectors. Here, H_f and W_f denote the height and width of the feature map, and C is the number of channels. This sequence is passed to the transformer encoder, which refines the features. The transformer decoder then uses learned object queries to attend to different regions of the image, refining them to focus on potential objects. Finally, these refined queries are passed to the output stage where two Feed Forward Networks (FFNs) decode each query into $\hat{y}_i = (\hat{c}_i, \hat{b}_i)$, which includes predicted classes and bounding boxes. These predictions are matched to the ground-truth objects using the Hungarian algorithm, which seeks to minimize the global matching cost, defined as

$$\hat{\sigma} = \arg\min_{\sigma} \sum_{i=1}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma_i}), \tag{1}$$

where $\hat{\sigma}$ denotes the optimal permutation of the predictions. The permutation σ, maps indices $\{1, 2, \ldots, N\}$ to the best assignment of predictions to ground truth objects. N is the total number of objects, and $\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma_i})$ represents the matching cost between the ground-truth object $y_i = (c_i, b_i)$ and the predicted object $\hat{y}_i = (\hat{c}_i, \hat{b}_i)$, defined as $\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma_i}) = \mathcal{L}_{\text{cls}}(c_i, \hat{c}_{\sigma_i}) + \mathbf{1}_{\{c_i\neq\emptyset\}}\mathcal{L}_{\text{bbox}}(b_i, \hat{b}_{\sigma_i})$ where $\mathcal{L}_{\text{cls}}(c_i, \hat{c}_{\sigma_i})$ denotes the classification loss between the predicted class $\hat{c}_{\sigma_i}$ and the ground truth class c_i, and $\mathbf{1}$ is the indicator function:

$$\mathbf{1}_{\{c_i\neq\emptyset\}} = \begin{cases} 1 & \text{if } c_i \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Finally, $\mathcal{L}_{\text{bbox}}(b_i, \hat{b}_{\sigma_i})$ denotes the bounding-box loss between the predicted box $\hat{b}_{\sigma_i}$ and the ground-truth box b_i. The predictions which are paired with a ground-truth object are called positive samples, while those which are not, are called negative samples. Once the optimal assignment $\hat{\sigma}$ is found, the overall detection loss $\mathcal{L}_{\text{det}}$ is computed as:

$$\mathcal{L}_{\text{det}}(y, \hat{y}_{\hat{\sigma}}) = \sum_{i=1}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\hat{\sigma}_i}). \tag{3}$$

In addition to using DETR's detection loss, our logit-level[1] distillation method is based on the approach of Chang *et al.* [26]. In this paper, the authors proposed to exploit both the positive and negative predictions in the distillation process. Let y^{T} and y^{S} denote the predictions of the Teacher and Student models, respectively. The loss for knowledge distillation on logits, $\mathcal{L}_{\text{logitsKD}}$, is computed separately, for positive and negative predictions as follows:

$$\mathcal{L}_{\text{logitsKD}}(\hat{y}^{\text{Tpos}}, \hat{y}^{\text{S}}_{\hat{\sigma}^{\text{pos}}}) = \sum_{i=1}^{N} \mathcal{L}_{\text{match}}(\hat{y}_i^{\text{Tpos}}, \hat{y}^{\text{S}}_{\hat{\sigma}_i^{\text{pos}}}), \tag{4}$$

$$\mathcal{L}_{\text{logitsKD}}(\hat{y}^{\text{Tneg}}, \hat{y}^{\text{S}}_{\hat{\sigma}^{\text{neg}}}) = \sum_{i=1}^{N} \mathcal{L}_{\text{match}}(\hat{y}_i^{\text{Tneg}}, \hat{y}^{\text{S}}_{\hat{\sigma}_i^{\text{neg}}}). \tag{5}$$

Since the Teacher's positive predictions are assumed to be closely related to the target, in [26] the authors used them as a knowledgeable pseudo ground truth. In equations (4) and (5), $\hat{y}_i^{\text{Tpos}}$ and $\hat{y}_i^{\text{Tneg}}$ represent the logits of the Teacher model's positive and negative predictions for the i-th instance, respectively. Similarly, $\hat{y}^{\text{S}}_{\hat{\sigma}_i^{\text{pos}}}$ and $\hat{y}^{\text{S}}_{\hat{\sigma}_i^{\text{neg}}}$ denote the Student model's predictions, which have been optimally matched to the Teacher model's positive and negative predictions. This alignment ensures that the Student not only learns from the correct detections, but also from instances where the Teacher confidently identifies the absence of objects.

[1]We focus here on the raw output values (*logits*) produced by the model before they are converted into probabilities by the softmax function.

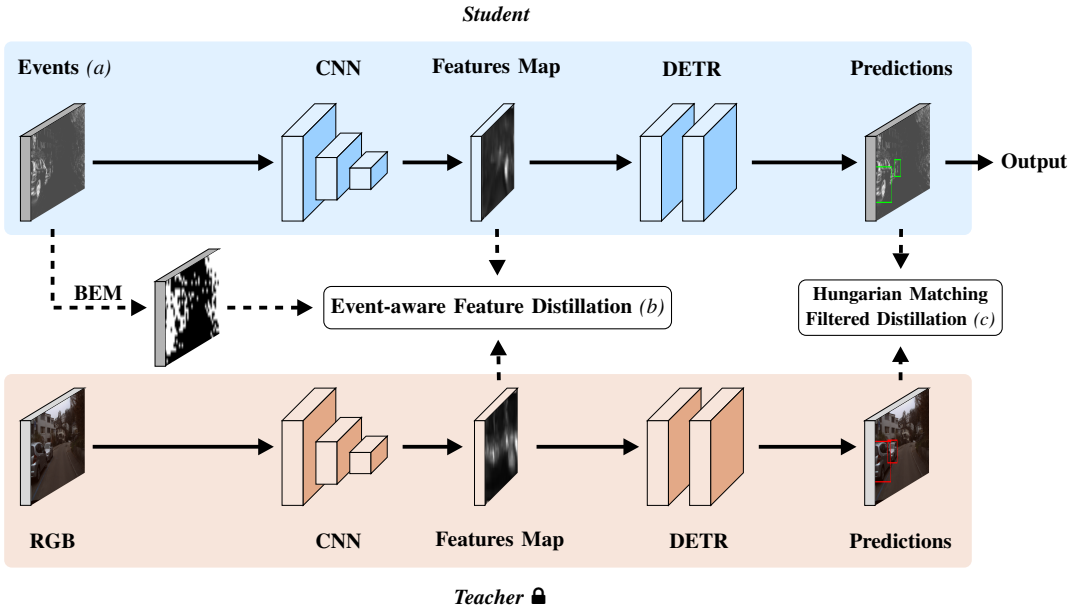

Fig. 2: Training pipeline of EA-DETR.

B. Event modeling

All the pixels in an event camera work independently and operate asynchronously. An event is generated when the brightness change exceeds a pre-defined threshold and it is represented by the 4-tuple, $\mathbf{e} = (x, y, t, p)$, where (x, y) are the spatial coordinates of the pixel, t is the timestamp, and $p \in \{-1, +1\}$ is the polarity, i.e. the sign of brightness change. In this paper, the events are pre-processed using a simple histogram representation (see Fig. 2(a)). For a given accumulation time T, a histogram $\mathbf{H}(x, y)$ is built by aggregating the events at each pixel location (x, y) over the time period T, $\mathbf{H}(x, y) = \sum_{t=t_0}^{t_0+T} \mathbf{e}(x, y, t)$ where $\mathbf{e}(x, y, t)$ represents the occurrence of an event at pixel (x, y) at time t, and t_0 is the current time. For the sake of simplicity, the polarity of the events is ignored.

C. Event-aware backbone distillation

In this section, we describe the backbone distillation process (see Fig. 2(b)). As mentioned in Sect. II-B, the disparity between the number of background and foreground pixels is problematic for knowledge distillation. We argue that event cameras can effectively get around this issue: in fact, thanks to the sparsity of events, the number of background pixels is significantly reduced, compared to an RGB image. Hence, we first generate a Binary Events Mask (BEM) $\mathbf{M}_{\text{events}} \in \{0, 1\}^{H \times W}$ from the event image tensor $\mathbf{H}_{\text{events}}$. This mask identifies the regions with events, where non-zero values in $\mathbf{H}_{\text{events}}$ become 1, and zero values remain 0. The binary mask $\mathbf{M}_{\text{events}}$ is resized to match the spatial dimension of the features of the Teacher model. To do this, we use a nearest-neighbor interpolation which maintains the original event locations by assigning the value of the nearest pixel. We define the backbone loss

$$\mathcal{L}_{\text{MSE}} = \frac{1}{M} \sum_{i=1}^{M} \left(\mathbf{F}_{\text{events}}^{(i)} - \mathbf{F}_{\text{RGB}}^{(i)}\right)^2, \tag{6}$$

which is the Mean Squared Error (MSE) between the Student model's feature map, $\mathbf{F}_{\text{events}}$, and the Teacher model's feature map, $\mathbf{F}_{\text{RGB}}$, with M the number of elements in the feature maps. In equation (6), $\mathbf{F}_{\text{events}}^{(i)}$ denotes the i-th element of the vectorized feature map generated by the Student model using event data, and $\mathbf{F}_{\text{RGB}}^{(i)}$ is the i-th element of the vectorized feature map produced by the Teacher model using RGB data. $\mathcal{L}_{\text{MSE}}$ measures the squared difference between the corresponding feature values, and pushes the Student to learn similar representations as the Teacher. To ensure that only the regions defined by the mask $\mathbf{M}_{\text{events}}$ contribute to the loss, we define

$$\mathcal{L}_{\text{MaskedBack}} = \frac{1}{\|\mathbf{M}_{\text{events}}\|_0} \sum_{i=1}^{N} \mathbf{M}_{\text{events}}^{(i)} \left(\mathbf{F}_{\text{events}}^{(i)} - \mathbf{F}_{\text{RGB}}^{(i)}\right)^2, \tag{7}$$

where $\| \cdot \|_0$ denotes the L_0 norm of a matrix, i.e. the number of non-zero elements of the matrix. This masked loss guarantees that only the relevant regions of the scene, as specified by the events, contribute to the training process.

D. Event-aware logits distillation

This section presents the logits distillation step (see Fig. 2(c)). The Hungarian-matching logits distillation technique proposed in [26] and described in Sect. II-B, has been modified to account for the sparse nature of event data. In fact, most of negative predictions from DETR, are empty. To assist the model in converging and to increase the convergence speed, we introduce an additional filter, $\mathbf{M}_{\text{neg}}$, which only selects negative detection boxes containing events. More precisely, equation (5) has been modified as

follows:

$$\mathcal{L}_{\text{MaskedLogs}}(\hat{y}^{\text{Tneg}}, \hat{y}^{\text{S}}_{\hat{\sigma}^{\text{neg}}}) = \sum_{i=1}^{N} \mathbb{I}(\mathbf{M}^{(i)}_{\text{neg}}) \mathcal{L}_{\text{match}}(\hat{y}^{\text{Tneg}}_{i}, \hat{y}^{\text{S}}_{\hat{\sigma}^{\text{neg}}_{i}}). \tag{8}$$

In $\mathcal{L}_{\text{MaskedLogs}}$, the indicator function $\mathbb{I}(\mathbf{M}^{(i)}_{\text{neg}})$ plays a critical role in determining whether the loss associated with a particular negative prediction $\hat{y}^{\text{Tneg}}_{i}$ is included in the final loss. More specifically, this function evaluates $\mathbf{M}_{\text{neg}}$ for the i-th prediction, i.e. $\mathbf{M}^{(i)}_{\text{neg}}$. If the corresponding detection box contains a sufficient number of events, exceeding a predefined threshold T_{mask}, then $\mathbb{I}(\mathbf{M}^{(i)}_{\text{neg}}) = 1$. In this way, the matching loss $\mathcal{L}_{\text{match}}$ for that prediction can contribute to the overall loss $\mathcal{L}_{\text{MaskedLogs}}$. Conversely, if the detection box is essentially empty, $\mathbb{I}(\mathbf{M}^{(i)}_{\text{neg}}) = 0$, and the matching loss is excluded from $\mathcal{L}_{\text{MaskedLogs}}$.

E. Overall loss

The overall loss function is the combination of the detection loss, logits knowledge distillation loss and feature knowledge distillation loss

$$\mathcal{L} = \mathcal{L}_{\text{det}} + \lambda_1 \mathcal{L}_{\text{MaskedBack}} + \lambda_2 \mathcal{L}_{\text{MaskedLogs}}, \tag{9}$$

where λ_1 and λ_2 are positive hyperparameters that can be used to weigh the contributions of masked-features and logits knowledge distillation losses, respectively.

IV. Experimental results

We evaluated EA-DETR on DSEC-DET, an extension of the DSEC dataset [17], which also includes ground-truth bounding boxes. To push our analysis a step further, we also created an extract of DSEC-DET, called Hard-DSEC-DET, which comprises challenging RGB images and provides ground-truth annotations. As a baseline, we trained a DETR model using event data *only* as input and no knowledge distillation.

A. Setup

Implementation details: For the definition of our model, we used PyTorch [33], while for training, we took advantage of PyTorch Lightning [34]. For the baseline DETR, we employed a ResNet101-DC5 backbone initialized with random weights. For the Teacher DETR, we employed the ResNet101-DC5 architecture with weights pre-trained on the COCO dataset. For knowledge distillation between the Teacher and the Student DETR, we utilized the optimal weights of the RGB model pre-trained on DSEC-DET. The two positive weights λ_1 and λ_2 in equation (9) were empirically set to 1, assuming equal importance of both terms. Similarly, the threshold T_{mask} for logits distillation was set to 1, which allowed us to consider Teacher boxes containing at least one event. Each training session was conducted with a batch size of 16 across 4 Tesla V100 GPUs. This setup required approximately 2 days to train DETR on DSEC-DET and 2 days for knowledge distillation. We trained our models over 50 epochs using AdamW optimizer [35].

Datasets: DSEC [17] is a driving scenario dataset which includes two event cameras (640×480 pixels) and two color cameras (1440×1080 pixels). It contains 53 sequences totaling 3193 seconds, split into 41 sequences (2634 seconds) for training and 12 sequences (559 seconds) for testing. Since ground-truth annotations for object detection are missing in the DSEC dataset, DSEC-DET has been recently created. It provides 60 sequences (70379 frames, 390118 bounding boxes) across 8 object classes: pedestrians, riders, cars, buses, trucks, bicycles, motorcycles, and trains. These annotations have become essential for benchmarking event-based detection models.

Hard-DSEC-DET, is a test subset, which focuses on challenging lighting conditions, like tunnel transitions. It allowed us to evaluate the robustness of EA-DETR in dynamic scenes, under extreme conditions, where RGB-only models work poorly. It currently consists of a single 16-second sequence (500 frames, 722 bounding boxes), with additional sequences planned for future inclusion.

Performance metrics: To evaluate the performance of EA-DETR, we employed the COCO's mean Average Precision (mAP) metric. It measures precision and recall across various intersection over union (IoU) thresholds, ranging from 0.5 (mAP50) to 0.95 (mAP95). This metric calculates the Average Precision (AP) at each IoU threshold by plotting the precision-recall curve and determining the area under it. The final mAP is the mean of these AP values, providing a comprehensive assessment of the model's accuracy in detecting objects with varying degrees of overlap. For example, the mAP50:95 score captures the model's performance by averaging precision across IoU thresholds ranging from 0.5 to 0.95 in increments of 0.05.

B. DSEC-DET: Comparison with the state-of-the-art

We compared EA-DETR with the state-of-the-art methods (see Table I). To the best of our knowledge, we are the first to perform RGB-event distillation on DSEC-DET, a direct comparison with the majority of existing methods, impossible. Since EA-DETR infers on event data alone, to be fair, we restrict our comparison to other event-only methods. To this end, we used the results from the ablation study of DAGr model [2], which relies solely on event data during inference and on our DETR baseline, as a benchmark. EA-DETR surpasses the DETR baseline, achieving an mAP50

Input type	Model	DSEC-DET		
		mAP50	mAP50:95	Time
RGB	Faster R-CNN [32]	35.4	18.2	58.2
	RetinaNet [36]	30.5	16.6	73
	CenterNet [37]	35.1	10.4	7.0
	YOLOv7-E6E [38]	31.5	18.2	27.8
	YOLOv5-L [39]	33.2	20.9	**4.4**
	Baseline DETR [16]	**50.6**	**27.7**	23.3
Events	Baseline DETR [16]	25.8	12.0	**23.3**
	DAGr [2]	–	14.0	–
	EA-DETR (ours)	**27.2**	**14.7**	**23.3**

TABLE I: Performance comparison of different state-of-the-art models on DSEC-DET. The fifth column of the table reports the average inference time per image, in milliseconds. The best values are shown in bold.

Fig. 3: Predictions on Hard-DSEC-DET: (first row, events) EA-DETR, (second row, color images) DETR-RGB baseline. The green bounding boxes correspond to the model's prediction, and the red bounding boxes to the ground truth.

Model	Hard-DSEC-DET				
	mAP50	mAP50:95	APS	APM	APL
Baseline DETR†	37.6	20.4	14.6	52.5	50.0
Baseline DETR*	31.5	14.6	**7.5**	46.0	23.5
EA-DETR (ours)	**31.6**	**15.3**	6.3	**49.5**	**47.2**

TABLE II: Performance comparison of our model, EA-DETR, on Hard-DSEC-DET. APS: Average Precision Small; APM: Average Precision Medium; APL: Average Precision Large. These acronyms refer to the detection of small, medium and large objects. Finally, the symbols "†" and "*" refer to the RGB and event baselines, respectively.

of 27.2 and an mAP50:95 of 14.7. However, a significant performance gap remains, when compared to the RGB-based models, with the RGB DETR baseline reaching an mAP50 of 50.6 and an mAP50:95 of 27.7. Other RGB models, like Faster R-CNN and YOLOv5-L, exhibit greater detection accuracy, showing that there is still room for further improvement. This gap persists because of limited specialized architectures for event cameras. EA-DETR's distillation mechanism and event-focused masks, are a first important step in this direction.

C. Hard-DSEC-DET: Performance comparison

Table II summarizes our results with Hard-DSEC-DET. We can see that EA-DETR outperforms the baseline DETR in the challenging mAP50:95 category, scoring 15.3 instead of 14.6. EA-DETR particularly excels at detecting medium and large objects, with significant improvements in APM, reaching 49.5, and APL, reaching 47.2, while the baseline achieves 46.0 and 23.5, respectively. However, it falls slightly behind in detecting small objects, scoring 6.3 (the best value is 7.5). A possible explanation for this phenomenon, is that DETR's aggressive spatial downsampling reduces small objects to very few feature tokens and BEM-based distillation concentrates on regions with high event activity (mostly corresponding to medium or large objects). This

Logits	Backbone	DSEC-DET				
		mAP50	mAP50:95	APS	APM	APL
✗	✗	**27.1**	12.8	2.6	17.6	41.5
✗	✓	25.8	13.7	2.8	18.8	42.3
✓	✓	26.6	**14.7**	**3.0**	**20.5**	**42.6**

TABLE III: Effect of distillation on DETR using RGB-event data and BEM (DSEC-DET dataset).

thus negatively impacts the detection of small-sized objects. However, this limitation is minor in an automotive contexts, where the detection of large objects has a priority due to collision risks. In addition to the previous quantitative analysis, Fig. 3 reports some qualitative graphical results. Green boxes show the model's predictions, and red boxes indicate ground truth. EA-DETR (top row) detects vehicles on the opposite lane more accurately than the DETR-RGB baseline, in low-light conditions, especially inside the tunnel.

D. Ablation studies

We finally evaluated the impact of our distillation mechanism through ablation studies on DSEC-DET. Table III reports results for logits and backbone distillation, applied separately or conjointly. Backbone distillation alone improves mAP50:95 by nearly one point, highlighting the importance of capturing event-specific features. Logits distillation also improves performance, though to a lesser extent. Their combination yields the best results, enhancing average precision for small, medium, and large objects (APS, APM, APL). supports the conclusion that the joint application of logits and backbone distillation is beneficial (cf. the convergence curves in the accompanying video).

V. Conclusion and Future Work

In this paper, we have proposed EA-DETR (Event-Aware DETR), a new transformer-based object detection model that optimally exploits RGB and event data via cross-modal knowledge distillation. Ablation studies on the DSEC-DET dataset have demonstrated the model's accuracy and robustness, especially in challenging dynamic conditions. In

future works, we plan to develop a spiking architecture that leverages our distillation method, in order to better exploit the sparse, asynchronous nature of events. This architecture efficiently integrates temporal and spatial features to enhance detection accuracy.

ACKNOWLEDGMENTS

This work was supported in part by the French National Research Agency through the CERBERE project (ANR-21-CE22-0006) and in part by the PHC STAR NasDroVie project (2023-2024). The authors had access to the HPC resources of the MatriCS platform at the Université de Picardie Jules Verne. The platform is co-financed by the European Union via the European Regional Development Fund (FEDER) and by the Hauts-de-France Regional Council, among others. They were also granted access to the HPC resources of IDRIS under the 2024-AD011015260 allocation from GENCI.

REFERENCES

[1] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, and A. Knoll. Event-Based Neuromorphic Vision for Autonomous Driving. *IEEE Signal Process. Mag.*, 37(4):34–49, 2020.

[2] D. Gehrig and D. Scaramuzza. Low-latency automotive vision with event cameras. *Nature*, 629(8014):1034–1040, 2024.

[3] E. Perot, P. de Tournemire, D. Nitti, J. Masci, and A. Sironi. Learning to Detect Objects with a 1 Megapixel Event Camera. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16639–16652. Curran Associates, Inc., 2020.

[4] M. Gehrig and D. Scaramuzza. Recurrent Vision Transformers for Object Detection With Event Cameras. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 13884–13893, 2023.

[5] L. Cordone, B. Miramond, and P. Thierion. Object Detection with Spiking Neural Networks on Automotive Event Data. In *Proc. Int. Joint Conf. Neur. Netw.*, pages 1–8, 2022.

[6] S. Barchid, J. Mennesson, J. Eshraghian, C. Djéraba, and M. Bennamoun. Spiking neural networks for frame-based and event-based single object localization. *Neurocomputing*, 559:126805, 2023.

[7] M. Nagaraj, C.M. Liyanagedera, and K. Roy. DOTIE - Detecting Objects through Temporal Isolation of Events using a Spiking Architecture. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 4858–4864, 2023.

[8] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, and Y. Andreopoulos. Graph-Based Spatio-Temporal Feature Learning for Neuromorphic Vision Sensing. *IEEE Trans. Image Process.*, 29:9084–9098, 2020.

[9] Y. Li, H. Zhou, B. Yang, Y. Zhang, and Z. Cui. Graph-Based Asynchronous Event Processing for Rapid Object Recognition. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 934–943, 2021.

[10] S. Schaefer, D. Gehrig, and D. Scaramuzza. AEGNN: Asynchronous Event-based Graph Neural Networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 12361–12371, 2022.

[11] G. Chen, F. Wang, W. Li, L. Hong, J. Conradt, J. Chen, Z. Zhang, Y. Lu, and A. Knoll. NeuroIV: Neuromorphic Vision Meets Intelligent Vehicle Towards Safe Driving With a New Database and Baseline Evaluations. *IEEE Trans. Intell. Transp. Syst.*, 23(2):1171–1183, 2022.

[12] Z. El Shair and S. Rawashdeh. High-temporal-resolution event-based vehicle detection and tracking. *Opt. Eng.*, 62:031209–031209, 2023.

[13] Z. Zhou, Z. Wu, R. Boutteau, F. Yang, C. Demonceaux, and D. Ginhac. RGB-Event Fusion for Moving Object Detection in Autonomous Driving. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 7808–7815, 2023.

[14] A. Tomy, A. Paigwar, K.S. Mann, A. Renzaglia, and C. Laugier. Fusing Event-based and RGB camera for Robust Object Detection in Adverse Conditions. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 933–939, 2022.

[15] A. Devulapally, M.F.F. Khan, S. Advani, and V. Narayanan. Multi-Modal Fusion of Event and RGB for Monocular Depth Estimation Using a Unified Transformer-based Architecture. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 2081–2089, 2024.

[16] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-End Object Detection with Transformers. In *Proc. Europ. Conf. Comput. Vis.*, pages 213–229, 2020.

[17] M. Gehrig, W. Aarents, D. Gehrig, and D. Scaramuzza. DSEC: A Stereo Event Camera Dataset for Driving Scenarios. *IEEE Rob. Autom. Lett.*, 6(3):4947–4954, 2021.

[18] Z. Liu, N. Yang, Y. Wang, Y. Li, X. Zhao, and F.-Y. Wang. Enhancing Traffic Object Detection in Variable Illumination With RGB-Event Fusion. *IEEE Trans. Intell. Transp. Syst.*, 25(12):20335–20350, 2024.

[19] G. Chen, W. Choi, X. Yu, T. Hanand, and M. Chandraker. Learning Efficient Object Detection Models with Knowledge Distillation. In *Proc. Neural Inf. Proces. Syst.*, volume 30, 2017.

[20] Q. Li, S. Jin, and J. Yan. Mimicking Very Efficient Network for Object Detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 6356–6364, 2017.

[21] T. Wang, L. Yuan, X. Zhang, and J. Feng. Distilling Object Detectors With Fine-Grained Feature Imitation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4933–4942, 2019.

[22] L. Zhang and K. Ma. Improve Object Detection with Feature-based Knowledge Distillation: Towards Accurate and Efficient Detectors. In *Proc. Int Conf. Learn. Repr.*, 2021.

[23] D. Meng, X. Chen, Z. Fan, G. Zeng, H. Li, Y. Yuan, L. Sun, and J. Wang. Conditional DETR for Fast Training Convergence. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 3651–3660, 2021.

[24] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *Proc. Int. Conf. Learn. Repr.*, 2021.

[25] Z. Gao, L. Wang, B. Han, and S. Guo. AdaMixer: A Fast-Converging Query-Based Object Detector. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 5364–5373, 2022.

[26] J. Chang, S. Wang, H.-M. Xu, Z. Chen, C. Yang, and F. Zhao. DETRDistill: A Universal Knowledge Distillation Framework for DETR-families. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 6898–6908, 2023.

[27] S.S. Kruthiventi, P. Sahay, and R. Biswal. Low-light pedestrian detection from RGB images using multi-modal knowledge distillation. In *Proc. IEEE Int. Conf. Image Process.*, pages 4207–4211, 2017.

[28] L. Li, A. Linger, M. Millhaeusler, V. Tsiminaki, Y. Li, and D. Dai. Object-centric cross-modal feature distillation for event-based object detection. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 15440–15447, 2024.

[29] X. Wang, H. Zhang, H. Yu, and X. Wan. Evlsd-ied: Event-based line segment detection with image-to-event distillation. *IEEE Trans. Instrum. Meas.*, 73(2530512):1–12, 2024.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin. Attention is All you Need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[31] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale. In *Proc. Int. Conf. Learn. Repr.*, 2021.

[32] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017.

[33] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *Proc. 31st Int. Conf. Neural Inf. Proc. Syst.*, 2017.

[34] K. Sawarkar. *Deep Learning with PyTorch Lightning: Swiftly build high-performance Artificial Intelligence (AI) models using Python*. Packt Publishing Ltd, 2022.

[35] I. Loshchilov and F. Hutter. Decoupled Weight Decay Regularization. In *Proc. Int. Conf. Learn. Repr.*, 2019.

[36] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal Loss for Dense Object Detection. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 2980–2988, 2017.

[37] X. Zhou, D. Wang, and P. Krähenbühl. Objects as Points. *arXiv preprint arXiv:1904.07850*, 2019.

[38] C.-Y. Wang, A. Bochkovskiy, and H.-Y.M. Liao. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 7464–7475, 2023.

[39] G. Jocher. YOLOv5 by Ultralytics, 2020.