LEOD: Label-Efficient Object Detection for Event Cameras

Ziyi Wu[1,2], Mathias Gehrig[3], Qing Lyu[1], Xudong Liu[1], Igor Gilitschenski[1,2]
[1]University of Toronto, [2]Vector Institute, [3] University of Zurich

Abstract

Object detection with event cameras benefits from the sensor's low latency and high dynamic range. However, it is costly to fully label event streams for supervised training due to their high temporal resolution. To reduce this cost, we present LEOD, the first method for label-efficient event-based detection. Our approach unifies weakly- and semi-supervised object detection with a self-training mechanism. We first utilize a detector pre-trained on limited labels to produce pseudo ground truth on unlabeled events. Then, the detector is re-trained with both real and generated labels. Leveraging the temporal consistency of events, we run bi-directional inference and apply tracking-based post-processing to enhance the quality of pseudo labels. To stabilize training against label noise, we further design a soft anchor assignment strategy. We introduce new experimental protocols to evaluate the task of label-efficient event-based detection on Gen1 and 1Mpx datasets. LEOD consistently outperforms supervised baselines across various labeling ratios. For example, on Gen1, it improves mAP by 8.6% and 7.8% for RVT-S trained with 1% and 2% labels. On 1Mpx, RVT-S with 10% labels even surpasses its fully-supervised counterpart using 100% labels. LEOD maintains its effectiveness even when all labeled data are available, reaching new state-of-the-art results. Finally, we show that our method readily scales to improve larger detectors as well. Code: https://github.com/Wuziyi616/LEOD.

1. Introduction

Object detection is key to scene understanding. It provides a compact representation of raw sensor measurements as semantically meaningful bounding boxes. Speed is crucial in object detection, especially in safety-critical applications such as self-driving. Recently, event cameras have gained significant interest in computer vision due to their low latency, low energy consumption, and high dynamic range [11]. Leveraging these benefits, event-based object detectors [14, 16, 30, 43, 50, 80] have been developed to complement conventional frame-based detectors. Despite tremendous progress, much of their success heavily relies
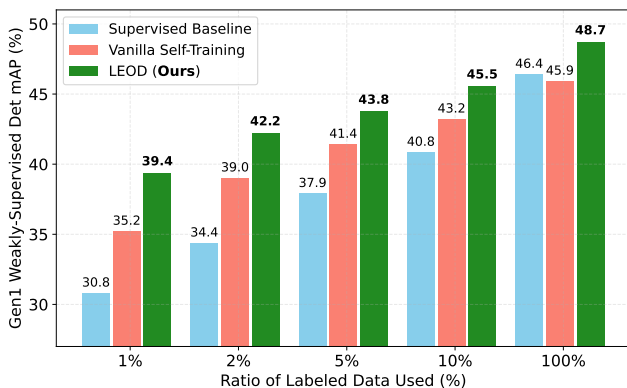

Figure 1. **Detection performance** of LEOD and baselines trained only on labeled events or conducting naive self-training. Under the weakly-supervised setting, our method consistently improves the RVT-S detector [14] across all labeling ratios on the Gen1 dataset.

on large datasets that are manually annotated. However, due to the high temporal resolution of event data, labeling objects at every timestamp is impractical. For example, Gen1 dataset [8] provides object labels at lower than 4 Hz. As a result, existing methods only train their models on labeled events, and discard the remaining unlabeled data. In contrast, we view this task as a weakly-supervised learning problem, which calls for detection methods that can leverage a mix of labeled and unlabeled events during training.

In this work, we address the challenge by proposing a **L**abel-efficient **E**vent-based **O**bject **D**etection (LEOD) framework. We consider two settings characterized by limited labels: **(i)** weakly-supervised, where object bounding boxes are sparsely labeled in all event streams, and **(ii)** semi-supervised, where some event streams have object boxes densely labeled while others remain fully unlabeled. Our approach unifies the two settings through a self-training paradigm. With limited labels, we first pre-train a detector and use it to generate pseudo annotations on unlabeled events. Then, we re-train the detector on a combination of real and pseudo labels. However, naïvely generated labels contain noise, making their direct use suboptimal.

To obtain high-quality pseudo labels, we exploit the temporal dimension of event data. Recent work [30] has shown the importance of temporal information in event-based detection using recurrent modules [53]. Additionally, offline label generation enables us to further refine predictions with

future information. To achieve this, we first introduce time-flip augmentation to events during training. As a result, we can ensemble model predictions on the original and reverse event streams via Test-Time Augmentation (TTA), leading to higher detection recall. In addition, we leverage tracking-based post-processing to eliminate temporally inconsistent objects, enhancing the precision of pseudo labels. Finally, we filter out low-confidence boxes with a score threshold.

A key challenge here is how to select a proper threshold. Instead of searching for the optimal value, we first filter with a low threshold to avoid missing objects. This inevitably introduces many false positives, which we address with a soft anchor assignment strategy in training. When computing the detection loss, we set another higher threshold and only treat pseudo boxes above that threshold as positive labels. For boxes with a lower detection score, we ignore the loss applied to their associated anchors. This strategy ensures that the model is only supervised with reliable background and foreground labels, while being tolerant to noisy labels. Ablation studies show that our method is insensitive to the two threshold values, easing the hyper-parameter tuning.

To test our method, we design new protocols for label-efficient event-based detection on Gen1 [8] and 1Mpx [43] datasets. For weakly-supervised object detection (WSOD), we uniformly sub-sample the labels over time to simulate sparse annotations. For semi-supervised object detection (SSOD), we directly choose some event sequences as fully unlabeled. Following 2D SSOD evaluation protocols [56, 67], we also have a fully labeled setting, where we show that pseudo labels can complement ground-truths.

In summary, this work makes four main contributions: **(i)** We introduce the task of label-efficient object detection to event vision, and design its experimental protocols. **(ii)** We propose LEOD, a unified framework for training event-based detectors with limited annotations. **(iii)** LEOD consistently outperforms baselines in various settings on two public detection datasets. **(iv)** Our method remains effective under the fully labeled setting and scales up to larger detectors, achieving new state-of-the-art performance.

2. Related Work

Object Detection with Event Cameras. Existing event-based detectors can be mainly categorized into two classes depending on whether they utilize the asynchronous nature of events. One line of work explores the sparsity of events, and employs Graph Neural Networks (GNNs) [13, 37, 50] or Spiking Neural Networks (SNNs) [7, 22, 73] for feature extraction. However, these approaches struggle with propagating information over long time horizons, which is crucial for detecting objects from events. Moreover, specialized hardware is required to achieve theoretical speed-ups of sparse networks, limiting their application in practice.

In another class of methods, events are converted to dense frame-like representations, followed by conventional networks for detection. Earlier works only consider event frames aggregated from a short time interval [5, 21, 23, 32]. This discards long-horizon history and makes it hard to detect objects under small relative motion as they trigger very few events. Recent methods thus introduce recurrent modules [17, 53] to enhance the memory of the detectors [14, 30, 43]. Further research focuses on better backbones [9, 59], inference speed [16], and event representations [80]. Because our primary goal is to study label-efficient learning for event-based detectors, we adopt the state-of-the-art approach RVT [14] as the base model.

Label-Efficient Learning in Event-based Vision. Due to a lack of large labeled datasets, there have been several works studying event-based algorithms with limited labels. Some papers focus on bridging frame-based and event-based vision. They either reconstruct natural images from events to apply traditional deep models [47, 48, 51, 57, 64], simulate events from videos to transfer the annotations [20, 40, 46, 79], or distill knowledge from trained frame-based models [19, 38, 58, 62, 70, 72]. However, these methods require either paired recordings of events and images or massive in-domain labeled images for training. Closer to ours are methods that only use event data [6, 27, 65, 78]. They conduct label-efficient learning on events with pre-trained frame-based models or self-supervised losses. Yet, none of them are designed for the detection task. Our work is the first attempt at label-efficient event-based object detection.

Label-Efficient Learning in Other Fields. Self-training based methods have been explored in tasks such as 2D image classification [28, 55, 66], object detection [56, 67, 75], and segmentation [1, 24, 42]. Our method is more related to label-efficient learning on videos [39, 54, 68] and 3D point cloud sequences [33, 45, 74] as these methods also exploit the temporal information of input data. For example, [68] utilizes optical flows to propagate single-frame labels to adjacent video frames. [45] and [33] train a teacher model on dense point clouds aggregated from a few timesteps. In contrast, we leverage a much longer temporal horizon by running the detector on the entire event stream in both directions. [74] also employs tracking-based post-processing to remove inconsistent boxes. We additionally perform tracking in both directions as a forward-backward consistency check, thus better leveraging the temporal information of event data. To tackle the noisy pseudo labels, [67] proposes to use the detection scores from the teacher model to weigh the loss. Instead, we design a soft anchor assignment strategy by ignoring the loss associated with unconfident boxes.

3. Method

This paper introduces a new task named label-efficient event-based object detection (formulated in Sec. 3.1). Our

algorithm adopts a two-stage self-training framework with reliable label selection (Sec. 3.2), where we leverage the temporal information of event data to obtain high-quality pseudo labels and suppress noisy predictions (Sec. 3.3).

3.1. Problem Formulation

Events data. Event cameras record brightness changes, and output a sequence of events $\mathcal{E} = \{e_i = (x_i, y_i, t_i, p_i)\}$. Each event e_i is parameterized by its pixel coordinate (x_i, y_i), timestamp t_i, and polarity $p_i \in \{-1, 1\}$. Modern event cameras run at sub-milliseconds and can produce millions of events per second [11].

Event-based Object Detection. As the object motion in a scene is usually much slower than the event generation speed, event-based object detectors are only applied and evaluated at a fixed time interval T [8, 43]. More specifically, given an event stream $\mathcal{E}$ capturing a set of objects $\mathcal{O} = \{o_j\}_{j=1}^M$, we aim at detecting the 2D bounding boxes of them with the semantic labels, $\mathcal{B} = \{b_j = (x_j, y_j, w_j, h_j, l_j, t_l)\}_{j=1}^M$. Each bounding box b_j is characterized by the location of its top-left corner (x_j, y_j), width w_j, height h_j, class label $l_j \in \{1, 2, ..., C\}$, and timestamp t_j. Here, C denotes the number of classes.

Event-based Object Detectors. We take RVT [14] for example as it serves as our base model. RVT is a synchronous detector that converts events in every time window Δt to a grid-like representation I. In the remaining part of the paper, we call I a *frame* and every Δt a *timestep*. RVT combines a Vision Transformer backbone [59] with a YOLOX detection head [12]. To extract temporal features, RVT introduces LSTM [17] cells in the backbone to fuse information over multiple timesteps. This is useful for detecting slow-moving objects as they only generate a few events. The YOLOX detection head in RVT is anchor-free, i.e., for each location on the feature map (anchor point), it predicts an objectness score $p_{\text{obj}} \in [0, 1]$, per-class IoU values $p_{\text{iou}} \in \mathbb{R}^C$, and offsets of the bounding box parameters $\Delta b = (\Delta x, \Delta y, \Delta w, \Delta h)$. The p_{iou} is trained to output the IoU value between the predicted box and the matched ground-truth box of that class. To obtain the final prediction, Non-Maximum Suppression (NMS) is applied to remove overlapping bounding boxes with low confidence.

During training, each ground-truth bounding box is matched to several anchor points for loss computation. Below we will denote anchor points as *anchors* for simplicity. Anchors matched with ground-truth are foreground, where all predicted values are supervised. For the remaining background anchors, only p_{obj} is trained to be 0.

Label-Efficient Event-based Object Detection. Fig. 2 shows our proposed detection settings with limited labels. *Weakly-supervised object detection (WSOD).* In the WSOD setting, all event streams are sparsely labeled. Moreover, labels assigned to adjacent frames offer fewer informa-
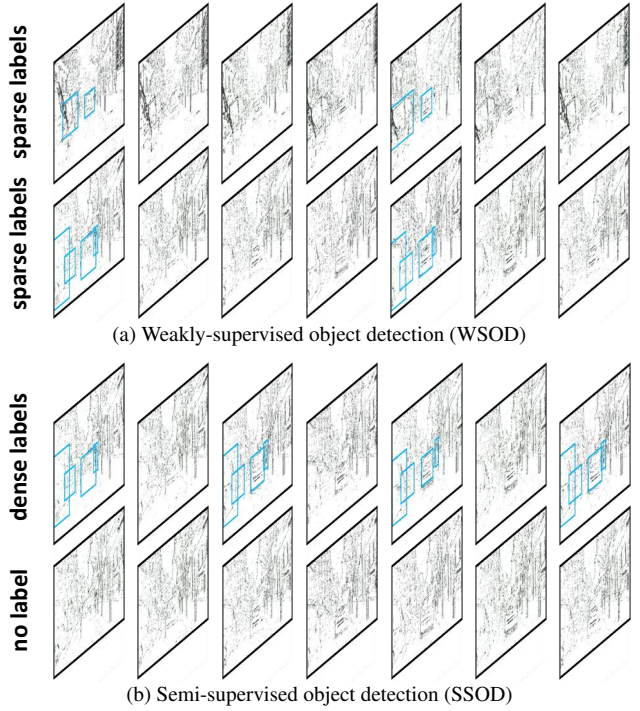

(a) Weakly-supervised object detection (WSOD)

(b) Semi-supervised object detection (SSOD)

Figure 2. **Illustration of two label-efficient event-based object detection settings**: (a) weakly-supervised where all event sequences are sparsely annotated, and (b) semi-supervised where some event sequences are densely annotated, and others are fully unlabeled. We visualize both positive and negative events in black.

tive training signals compared to those distributed across frames [68]. Therefore, it is reasonable to label object boxes in a long event sequence uniformly and sparsely.

Semi-supervised object detection (SSOD). In the SSOD setting, some event sequences are densely labeled, while others are fully unlabeled. This is also practical when people continue to collect data into already annotated datasets. Since capturing event sequences is much easier than labeling them, an algorithm that can consistently improve model performance with raw events is highly useful.

To evaluate the label-efficient learning performance, we take existing event-based object detection datasets [8, 43] and sample a small portion of frames (WSOD) or sequences (SSOD) as labeled data. The rest of the training data are used as an unlabeled set following previous works [34, 56, 67]. We also have a fully labeled setting where all labels are available. Since the original event streams are annotated at a larger time interval than Δt, we can still create pseudo labels on unlabeled timesteps to improve the performance. See Appendix D for further discussions on the two settings.

3.2. LEOD: A Self-Training Framework

As shown in Fig. 3, the overall pipeline of LEOD follows a student-teacher pseudo-labeling paradigm, which is applicable to both, the WSOD and the SSOD settings. We first pre-train a detector on labeled data using regular detection loss until convergence. Then, we employ it to annotate unlabeled frames. To leverage the benefit of offline prediction,
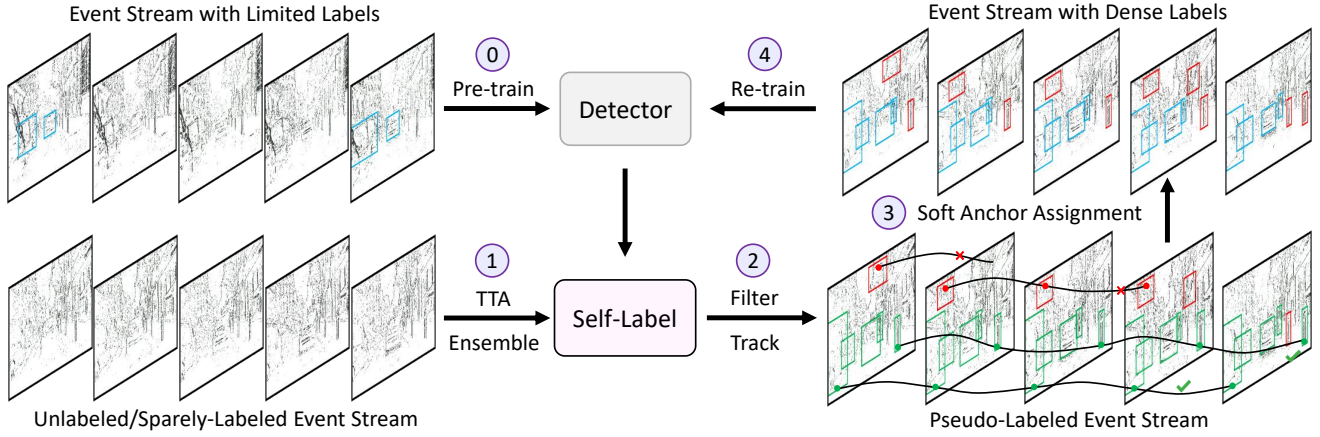
3

Figure 3. **Overview of our LEOD pipeline.** ⓪ We first pre-train an event-based object detector on event streams with limited labels. ① To leverage the temporal information, we apply time-flip Test-Time Augmentation (TTA) to unlabeled event streams and ensemble the model predictions. ② We then apply forward and backward tracking to identify temporally inconsistent bounding boxes, i.e., boxes associated with short tracks. ③ To handle noisy labels, a soft anchor assignment strategy is designed to ignore detection loss on unconfident pseudo labels (red boxes). ④ We can boost the model performance by self-training on reliable pseudo labels (blue boxes) and repeating ① – ④.

we apply temporal flip to get event streams in both directions, and aggregate the detection results on them. Since the teacher model is trained on limited data, it will be uncertain on hard examples. We thus threshold the boxes with a small value to keep more detected objects. To remove false positive boxes, we build upon the temporal persistency prior of objects and apply tracking-based post-processing. However, there might still be inaccurate labels due to the low confidence threshold we use, and directly training the detector on them will lead to suboptimal results. Inspired by prior works on noise-robust learning [29, 63], we design a soft anchor assignment strategy to selectively supervise the model with pseudo labels. Finally, we can use the re-trained detector as the teacher model to initialize the next round of self-training. The described process can be repeated for multiple rounds to further boost the model performance.

Comparison to online pseudo-labeling. In previous label-efficient object detection works [34, 61, 67, 71], the teacher model is jointly trained with a student model. In each training step, the teacher predicts bounding boxes on unlabeled data in a batch for student training. This online paradigm is also applicable to our setting. However, the teacher model will only see short event streams loaded in a batch. For example, on Gen1 [8], our training sequence length accounts for a duration of 1 second, while a car can stop for more than 10 seconds in real-world traffic and thus trigger no events. Pseudo-labeling on short event streams will inevitably miss these objects. Therefore, we adopt our two-stage offline label generation paradigm to retain full temporal information.

3.3. Towards High-Quality Pseudo Labeling

In this section, we introduce each key component in our LEOD framework to achieve high-quality pseudo labels.

Test-Time Augmentation (TTA). When deployed in the real world, event-based detectors are expected to run in real-time, i.e., they only take in events triggered before t to de-
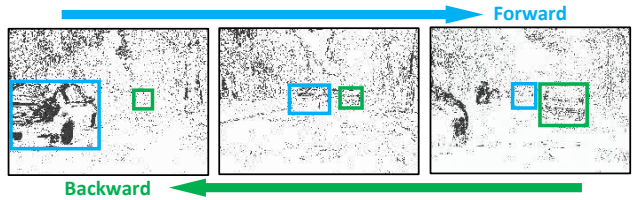

Figure 4. **Illustration of the time-flip TTA** which enhances our robustness against different object motions. Forward helps detect receding objects, while Backward helps with approaching objects.

tect objects at t. Instead, in our offline label generation process, we can use future information to refine predictions at the current timestep. As shown in Fig. 4, we run the detector on both the original and reversed event streams, enabling us to detect objects with different movements. We also apply a horizontal-flip TTA to further improve the detection.

Filtering and Tracking. TTA helps us detect more objects (higher recall), yet it also leads to false positives (lower precision). Previous works simply use a threshold to filter out boxes with low confidence [61, 67]. However, as shown in Fig. 5 (a) and (b), there is a trade-off between precision and recall, making it hard to find the optimal threshold. We opt to first filter with a low threshold τ_{hard} to avoid missing objects, followed by tracking-based post-processing to remove temporally inconsistent boxes. We follow the tracking-by-detection paradigm [2] to build tracks by linking detection boxes between frames. Each box b will be associated with a track $s_k = \{(b, v_x, v_y)_t, k, n, q\}$, where (v_x, v_y) is the estimated velocity under a linear motion assumption, k is the ID, n is its length so far, and $q \in [0, 1]$ is the current score. In the first frame, we initialize each box as a track. For every coming frame, we first predict the positions of existing tracks and associate them with boxes at that frame via greedy matching over pairwise IoUs. Then, we decay the score q of unmatched tracks and initialize unmatched

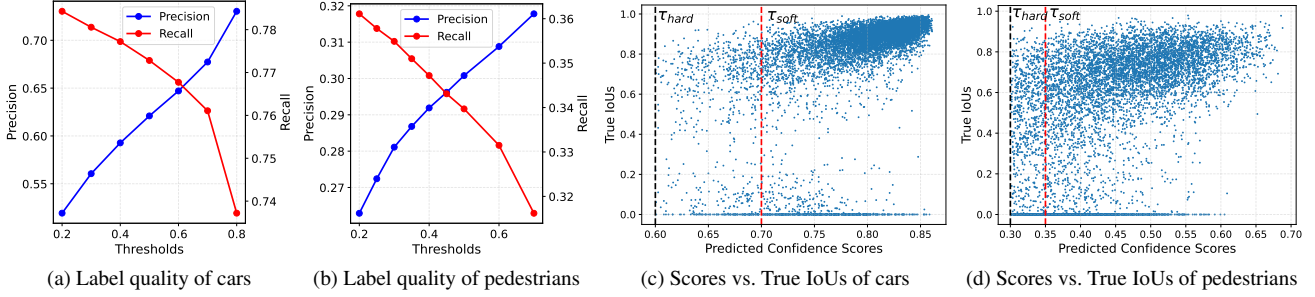| (a) Label quality of cars | (b) Label quality of pedestrians | (c) Scores vs. True IoUs of cars | (d) Scores vs. True IoUs of pedestrians |

Figure 5. **Analysis on confidence thresholds.** We randomly sample 10,000 predicted boxes from RVT-S pre-trained on 5% of Gen1 labels. We plot the pseudo labels' precision and recall of (a) cars and (b) pedestrians. In (c) and (d), we show each box's predicted confidence scores and its true IoU with ground-truth boxes. τ_{hard} is the threshold for initial filtering, and τ_{soft} is used in soft anchor assignment.

boxes as new tracks. Finally, tracks with low scores will be deleted. See Appendix A.1 for implementation details.

Similar to TTA, we apply tracking in both directions, and only remove a box if the length of its associated track is shorter than a threshold, T_{trk}, in both cases. However, predictions on hard examples may also be inconsistent, as the pre-trained model has limited capacity. Instead of suppressing removed boxes as background, we ignore them during loss computation as will be described later. Also, for long tracks, we inpaint boxes with linear motion at unmatched timesteps and ignore training losses on them too.

Soft Anchor Assignment and Re-training. We can now re-train a detector on the ground-truth labels and the pseudo labels with the original detection loss. However, as shown in Fig. 5 (c) and (d), there are still low-quality boxes after post-processing. To handle noisy labels, we utilize a soft anchor assignment strategy to selectively supervise the model training. We first identify a set of uncertain labels including boxes belonging to short tracks, inpainted from long tracks, and those with detection scores lower than a threshold τ_{soft}. Then, at each training step, we ignore the loss applied to anchors associated with these uncertain boxes, i.e., we do not supervise those anchors and allow them to discover new instances. This design is inspired by the anchor assignment in anchor-based detectors [31, 49], where two thresholds are used to determine foreground or background anchor boxes, and the anchors in between are ignored in loss computation. As we will show in ablation studies, soft anchor assignment makes our method less sensitive to hyper-parameters.

Despite training on noisy labels, the model learns to refine the labels and detect new objects. Thus, we do an additional round of self-training to further improve the results.

4. Experiments

Sec. 4.2 shows that LEOD outperforms baselines significantly in the low-label regime. In the fully labeled setting, we surpass previous state-of-the-art (Sec. 4.3). Our ablations show the contribution of each component in Sec. 4.4.

4.1. Experimental Setup

Datasets. We adopt Gen1 [8] and 1Mpx [43] datasets that feature various driving scenarios. Gen1 consists of 39 hours

of recordings with a 304×240 resolution event camera [44]. It provides bounding box annotations of cars and pedestrians at 1, 2, or 4 Hz. 1Mpx is recorded with a higher 720×1280 resolution event camera [10]. It contains around 15 hours of data collected over several months at day and night, and provides labels for cars, pedestrians, and two-wheelers at 30 or 60 Hz. Following previous works [14, 30], we remove ground-truth boxes that are too small during evaluation on both datasets, and half the events' resolution to 360×640 on 1Mpx.

Evaluation Protocol. Mean average precision (mAP) is adopted as the main performance metric. We choose 1%, 2%, 5%, and 10% as the labeling ratio following prior works [56, 67]. In the weakly-supervised object detection (WSOD) setting, labels in all event streams are uniformly sub-sampled. In the semi-supervised object detection (SSOD) setting, we keep a small portion of event streams unchanged, while setting other event sequences as fully unlabeled. For the same labeling ratio, the amounts of available labels in WSOD and SSOD are roughly the same. Finally, all labels are provided in the fully labeled setting.

Detector Training Details. We adopt the state-of-the-art event-based detector RVT [14] as our base model. Due to limited computation resources, we mainly experiment with RVT-S, while we show that LEOD also scales to the largest RVT-B variant in Sec. 4.4. Most of the configurations are the same as we build upon their open-source codebase. Here we only highlight our modifications. In order to apply the time-flip TTA, we train with an additional time-flip data augmentation. When re-training on pseudo labels, we initialize RVT from scratch and use the Adam optimizer [25] with a peak learning rate of 5×10^{-4} to train for 150k iterations. Please refer to Appendix A.2 for more details.

Pseudo-Labeling Details. Inspired by prior works [52, 61], we set different thresholds for each category. To simplify parameter tuning, we follow two rules: **(i)** pedestrians and two-wheelers share the same values, which are half of cars' values, **(ii)** for cars, the soft threshold τ_{soft} equals to the hard threshold $\tau_{\text{hard}} + 0.1$, while for pedestrians and two-wheelers, we use $\tau_{\text{soft}} = \tau_{\text{hard}} + 0.05$. In both settings and both datasets, we choose the same set of hyper-parameters: $\tau_{\text{hard}} = 0.6$ for cars and $\tau_{\text{hard}} = 0.3$ for pedestrians. Only in
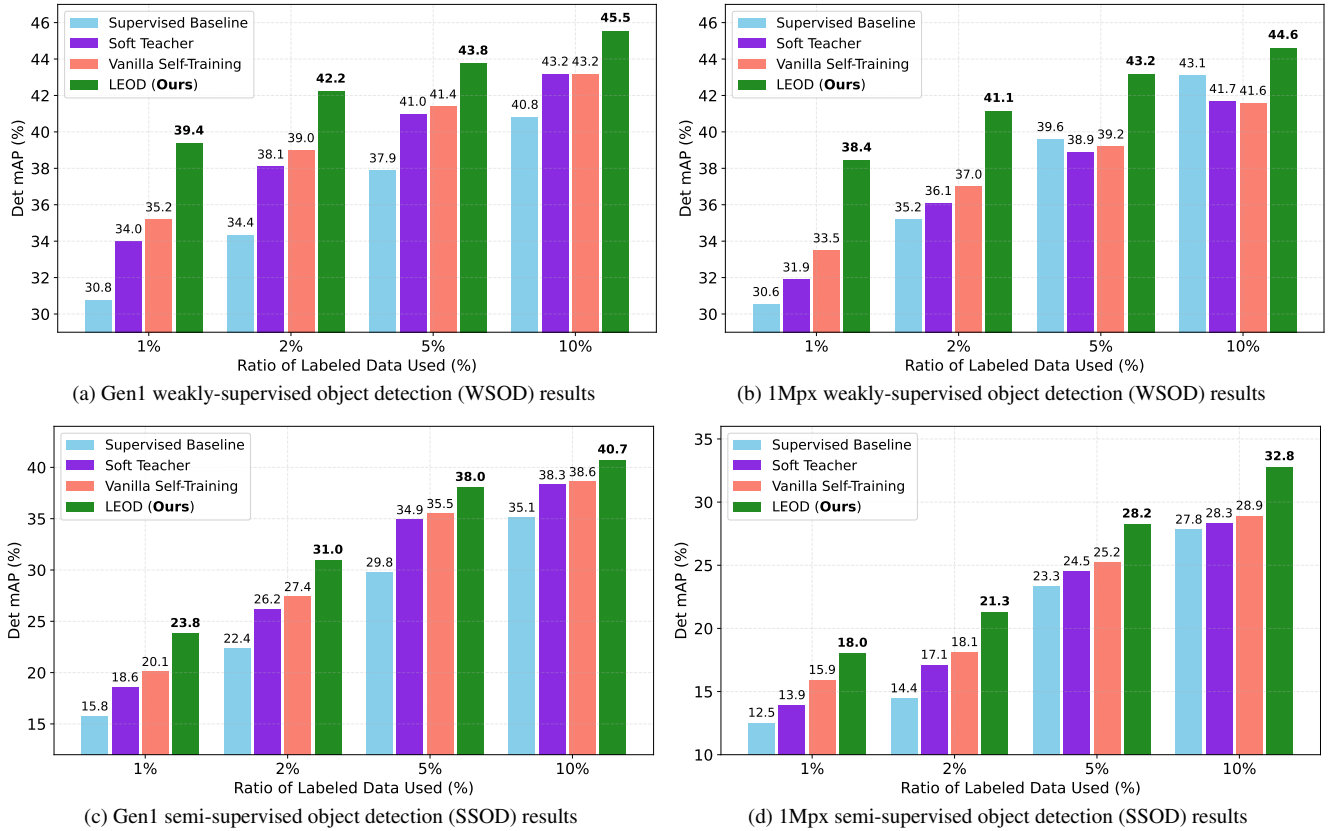
Figure 6. **Results on label-efficient learning** using different ratios of labeled data. We compare LEOD with baselines on Gen1 and 1Mpx datasets under WSOD and SSOD using the RVT-S detector. All results are averaged over three runs.

1Mpx WSOD, we set $\tau_{\text{hard}} = 0.5$ for pedestrians and two-wheelers to handle excessive false positives. The minimum track length T_{trk} is set to 6 in all experiments.

Baselines. We compare with a *Supervised Baseline* trained only on available labels. Since we are the first work to consider this task, we design two other baselines: *1) Vanilla Self-Training:* trains on pseudo labels without TTA, tracking, and soft anchor assignment; *2) Soft Teacher:* adopts the online student-teacher paradigm from a representative 2D SSOD method [67]. We enable soft anchor assignment in *Soft Teacher*, while TTA and tracking are not applicable as the online event sequence length is too short. We tune baselines' hyper-parameters to be optimal in each setting. We also tried a state-of-the-art 2D SSOD method designed for anchor-free detectors [35] as YOLOX is anchor-free, but we did not observe clear improvements over Soft Teacher.

4.2. Label-Efficient Results

We first compare our method with baselines in the low-labeled data regime. The overall results are shown in Fig. 6.
WSOD. Fig. 6 (a) and (b) present the weakly-supervised results. On Gen1, LEOD improves the mAP of Supervised Baseline by a large margin across all labeling ratios. Using 10% labels, we achieve an mAP of 45.5%, which is only 1% lower than RVT-S trained on 100% labels. For pseudo-labeling baselines, Vanilla Self-Training outperforms Soft Teacher in most cases, validating our choice of offline la-

bel generation. In addition, LEOD consistently outperforms them by more than 2%, indicating the higher quality of our pseudo labels. We observe a similar trend on 1Mpx, where our approach scores the highest mAP in all cases. Notably, the two pseudo-labeling baselines perform worse than pretrained RVT-S on 5% and 10% labels, which proves the importance of screening reliable labels. Finally, LEOD with 10% labels (44.6%) outperforms RVT-S trained on all labels (44.1%), showing the great potential of unlabeled data.
SSOD. Fig. 6 (c) and (d) present the semi-supervised results. Using the same amount of labels, models trained under SSOD are generally worse than WSOD. This indicates that given a limited budget, we should sparsely label as many event streams as possible instead of densely labeling a few sequences. Nevertheless, LEOD still outperforms baselines by more than 2% mAP over all labeling ratios on both datasets. Our results offer a promising direction of boosting performance with fully unlabeled event data.
Qualitative Results. Fig. 7 visualizes some detection results of RVT-S trained with 10% labels under Gen1 WSOD setting. The supervised baseline can only detect objects that trigger lots of events due to its limited capacity. Models using vanilla self-training detect more objects, but also produce numerous false positives. With our pseudo-labeling pipeline, LEOD trained models are able to handle various hard examples. For example, it discovers a car that was initially missed in the ground-truth annotation in Fig. 7 (d).
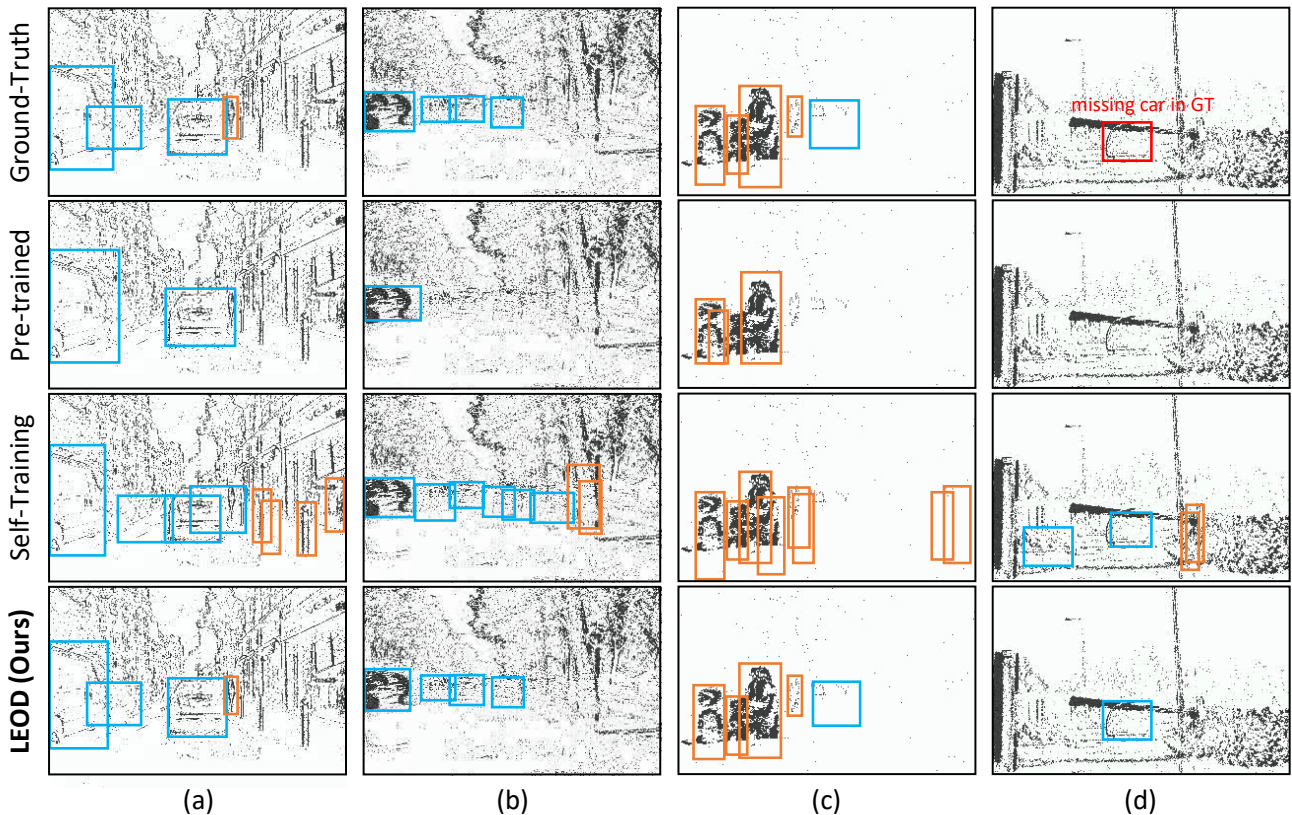
Figure 7. **Detection on Gen1.** We show (a) a common driving scene, (b) cars with small relative movement thus few events, (c) crowded pedestrians, and (d) LEOD discovers a missing object in ground-truth annotations due to the occlusion of a barrier gate. Car and pedestrian boxes are colored in blue and orange, respectively. Pre-trained and Self-Training stand for *Supervised Baseline* and *Vanilla Self-Training*.

4.3. Fully-Labeled Results

Since the original labeling frequency on both datasets is lower than the frame rate of RVT-S (20 Hz), we can still create pseudo labels on unlabeled frames to improve the fully supervised model performance. We compare with state-of-the-art event-based object detectors trained on all labeled data in Tab. 1. LEOD improves over RVT-S by 2.2% and 2.6% on Gen1 and 1Mpx, respectively. On Gen1, our method achieves new state-of-the-art among models not using pre-trained weights. This indicates that LEOD is consistently effective even with 100% labels. In terms of runtime and model size, since our approach does not introduce new modules to the base model, we are as efficient as RVT-S.
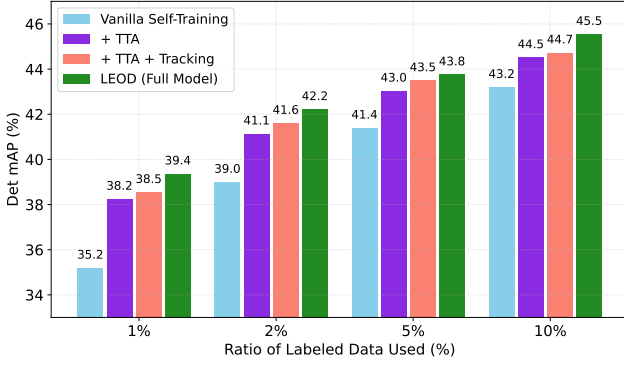
4.4. Ablation Studies

Larger Base Model. We evaluate our label-efficient learning framework on a stronger event-based detector RVT-B, which uses a larger ViT backbone compared to RVT-S. Tab. 2 presents the result in the Gen1 WSOD setting. With more parameters, RVT-B pre-trained on limited labels already outperforms RVT-S. Still, LEOD is able to improve the detection result by a sizeable margin across all labeling ratios. With 100% labels, our method achieves an mAP of 50.2%, which is competitive with ERGO-12 using a large-scale pre-trained Swin Transformer V2 backbone [36]. No-

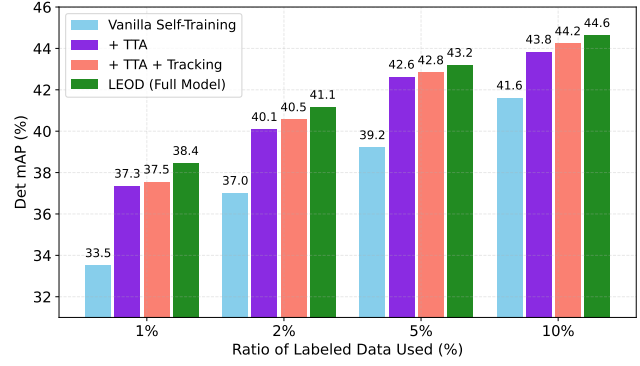| Method	Gen1		1Mpx		Size (M)
	mAP (%)	Time (ms)	mAP (%)	Time (ms)	
RED [43]	40.0	16.7	43.0	39.3	24.1
ASTMNet [30]	46.7	35.6	**48.3**	72.3	> 100
HMNet-L3 [16]	47.1	7.9*	-	-	33.2
RVT-B	47.2	10.2	47.4	11.9	18.5
RVT-S	46.5	9.5	44.1	10.1	9.9
LEOD-RVT-S	**48.7**	9.5	46.7	10.1	9.9
ERGO-12 [80]	50.4	77.2	40.6	101.1	59.6

Table 1. **Detection results using all available labels.** Baseline runtimes and model sizes are obtained from [14]. For HMNet, we use the best-performing L3 variant. * Its runtime is computed using a V100 GPU, while T4 GPUs which are slower than V100 are used in the other cases. ERGO is grayed as it is the only method using pre-trained models (Swin Transformer V2 [36]).

tably, LEOD brings larger absolute improvements on RVT-B compared to RVT-S, proving that our framework steadily scales up to enhance larger and stronger detectors.

Effect of Each Component. Fig. 8 shows the model performance with different components on Gen1 and 1Mpx. TTA significantly improves the mAP as it increases the recall in the generated pseudo labels using future information. Leveraging the temporal persistency of objects, tracking-based post-processing leads to a further gain. Finally, with soft anchor assignment, only reliable foreground and background labels are selected, easing the model training.

7

(a) Gen1 weakly-supervised object detection (WSOD) results

(b) 1Mpx weakly-supervised object detection (WSOD) results

Figure 8. **Ablation study of each component in LEOD**. We report the WSOD result with RVT-S as the base detector on both datasets. Starting from Vanilla Self-Training, we gradually add TTA, tracking-based post-processing, and soft anchor assignment.

Method	1%	2%	5%	10%	100%
RVT-S	30.8	34.4	37.9	40.8	46.5
LEOD-RVT-S	39.4	42.2	43.8	45.6	48.7
Absolute Improvement	+8.6	+7.8	+5.9	+4.8	+2.2
RVT-B	31.6	34.8	38.3	41.0	47.6*
LEOD-RVT-B	**40.0**	**42.9**	**45.3**	**46.6**	**50.2**
Absolute Improvement	+8.4	+8.1	+7.0	+5.6	+2.6

Table 2. **Gen1 WSOD mAPs (%) with two RVT variants.** * our reproduced RVT-B using 100% labels result is better than [14].

Rounds	1%	2%	5%	10%	100%	P. (1%)	P. (2%)	P. (5%)	P. (10%)
1	38.1	41.1	43.1	45.3	48.5	0.65	0.69	0.74	0.79
2	39.4	42.2	**43.8**	**45.6**	**48.7**	0.72	**0.75**	**0.77**	**0.81**
3	**39.5**	42.2	43.6	45.4	48.6	0.72	0.74	0.74	0.76

Table 3. **Number of self-training rounds used in LEOD.** We report the mAP (%) result in Gen1 WSOD using RVT-S. We also compute the precision (P.) of pseudo labels that are used to train the models on that row, which is a good indicator for stop training.

Self-Training Rounds. Tab. 3 left presents the results of multi-round self-training. Detectors after the first round of self-training are significantly better than pre-trained models, and thus generate higher quality pseudo labels. Therefore, a second round of training leads to consistent gains. However, due to error accumulations, a third round of training may result in worse models. To determine when to stop training, we empirically find that the precision of pseudo labels serves as a good indicator. In Tab. 3 right, we compute the precision of predicted boxes on labeled frames (skipped labels are not used to prevent information leakage). The precision consistently improves after round 1, but starts to decrease in some cases after round 2. Indeed, the mAP also drops after training on those labels with lower precision.

Confidence Thresholds. We analyze the effects of hard and soft thresholds (τ_{hard}, τ_{soft}) in Tab. 4. Prior works [61, 67] often use a high threshold of 0.9, while we observe clearly lower mAPs when $\tau_{hard} > 0.7$. With a lower τ_{hard}, we retain most of the detected objects, and suppress noisy labels with τ_{soft}. LEOD achieves similar results using several sets of hyper-parameters, showing our robustness to them. See Appendix B for more analysis on the filtering thresholds.

$(\tau_{hard, car}, \tau_{soft, car}, \tau_{soft, ped})$	1%	2%	5%	10%
$(0.5, 0.6, 0.3)$	38.9	41.8	43.5	45.4
$(\mathbf{0.6, 0.7, 0.35})$	**39.4**	42.2	**43.8**	45.5
$(0.6, 0.7, 0.4)$	39.3	42.0	43.7	**45.6**
$(0.6, 0.8, 0.4)$	39.3	**42.3**	43.6	45.5
$(0.7, 0.8, 0.4)$	39.0	42.0	43.2	44.9
$(0.8, -, -)^*$	38.4	41.4	42.7	44.2

Table 4. **Ablation of hard and soft thresholds used in soft anchor assignment.** We keep $\tau_{hard, ped} = \tau_{hard, car}/2$. Ped stands for pedestrian. We report the mAP (%) result in Gen1 WSOD using RVT-S. * indicates not using soft anchor assignment.

5. Conclusion

We present LEOD, the first algorithm for label-efficient event-based object detection. To leverage unlabeled data, we adopt the self-training framework with reliable label selection. Several techniques are introduced to improve labeling quality. Extensive experiments on Gen1 and 1Mpx datasets showcase the superiority of our method over baselines in all the settings.

Limitations and Future Work. Following common practice, we only conduct intra-dataset experiments, i.e., training on data gathered using the same protocol. Recent works [3, 26, 41] have shown that training large models over multiple datasets leads to excellent performance and generalization. Since LEOD benefits from unlabeled data, we can also train it jointly on more datasets that involve real-world multi-object event sequences [4, 15, 77]. We discuss some failure cases of our pseudo-labeling method in Appendix C.

References

[1] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *CVPR*, 2020. 2

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, 2016. 4, 12

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020. 8

[4] Kenneth Chaney, Fernando Cladera, Ziyun Wang, Anthony Bisulco, M. Ani Hsieh, Christopher Korpela, Vijay Kumar, Camillo J. Taylor, and Kostas Daniilidis. M3ed: Multi-robot, multi-sensor, multi-environment event dataset. In *CVPRW*, 2023. 8

[5] Nicholas FY Chen. Pseudo-labels for supervised learning on dynamic vision sensor data, applied to object detection under ego-motion. In *CVPRW*, 2018. 2

[6] Hoonhee Cho, Hyeonseong Kim, Yujeong Chae, and Kuk-Jin Yoon. Label-free event-based object recognition via joint learning with image reconstruction from events. In *ICCV*, 2023. 2

[7] Loïc Cordone, Benoît Miramond, and Philippe Thierion. Object detection with spiking neural networks on automotive event data. In *IJCNN*, 2022. 2

[8] 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. 1, 2, 3, 4, 5, 12

[9] 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. In *ICLR*, 2021. 2

[10] Thomas Finateu, Atsumi Niwa, Daniel Matolin, Koya Tsuchimoto, Andrea Mascheroni, Etienne Reynaud, Pooria Mostafalu, Frederick Brady, Ludovic Chotard, Florian LeGoff, et al. 5.10 a 1280× 720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86 μm pixels, 1.066 geps readout, programmable event-rate controller and compressive data-formatting pipeline. In *2020 IEEE International Solid-State Circuits Conference*, 2020. 5

[11] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *TPAMI*, 2020. 1, 3

[12] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 3, 12

[13] Daniel Gehrig and Davide Scaramuzza. Pushing the limits of asynchronous graph-based object detection with event cameras. *arXiv preprint arXiv:2211.12324*, 2022. 2

[14] Mathias Gehrig and Davide Scaramuzza. Recurrent vision transformers for object detection with event cameras. In *CVPR*, 2023. 1, 2, 3, 5, 7, 8, 12

[15] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *RA-L*, 2021. 8

[16] Ryuhei Hamaguchi, Yasutaka Furukawa, Masaki Onishi, and Ken Sakurada. Hierarchical neural memory network for low latency event processing. In *CVPR*, 2023. 1, 2, 7

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997. 2, 3

[18] Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. *NeurIPS*, 2017. 12

[19] Yuhuang Hu, Tobi Delbruck, and Shih-Chii Liu. Learning to exploit multiple vision modalities by using grafted networks. In *ECCV*, 2020. 2

[20] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. v2e: From video frames to realistic dvs events. In *CVPR*, 2021. 2

[21] Massimiliano Iacono, Stefan Weber, Arren Glover, and Chiara Bartolozzi. Towards event-driven object detection with off-the-shelf deep learning. In *IROS*, 2018. 2

[22] Mingcheng Ji, Ziling Wang, Rui Yan, Qingjie Liu, Shu Xu, and Huajin Tang. Sctn: Event-based object tracking with energy-efficient deep convolutional spiking neural networks. *Frontiers in Neuroscience*, 2023. 2

[23] Zhuangyi Jiang, Pengfei Xia, Kai Huang, Walter Stechele, Guang Chen, Zhenshan Bing, and Alois Knoll. Mixed frame-/event-driven fast pedestrian detection. In *ICRA*, 2019. 2

[24] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, 2017. 2

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[26] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 8

[27] Simon Klenk, David Bonello, Lukas Koestler, and Daniel Cremers. Masked event modeling: Self-supervised pretraining for event cameras. *arXiv preprint arXiv:2212.10368*, 2022. 2

[28] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICMLW*, 2013. 2

[29] Hengduo Li, Zuxuan Wu, Chen Zhu, Caiming Xiong, Richard Socher, and Larry S Davis. Learning from noisy anchors for one-stage object detection. In *CVPR*, 2020. 4

[30] Jianing Li, Jia Li, Lin Zhu, Xijie Xiang, Tiejun Huang, and Yonghong Tian. Asynchronous spatio-temporal memory network for continuous event-based object detection. *TIP*, 2022. 1, 2, 5, 7

[31] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 5

[32] Bingde Liu, Chang Xu, Wen Yang, Huai Yu, and Lei Yu. Motion robust high-speed light-weighted object detection with event camera. *IEEE Transactions on Instrumentation and Measurement*, 2023. 2

[33] Minghua Liu, Yin Zhou, Charles R Qi, Boqing Gong, Hao Su, and Dragomir Anguelov. Less: Label-efficient semantic segmentation for lidar point clouds. In *ECCV*, 2022. 2, 14

[34] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. In *ICLR*, 2020. 3, 4

[35] Yen-Cheng Liu, Chih-Yao Ma, and Zsolt Kira. Unbiased teacher v2: Semi-supervised object detection for anchor-free and anchor-based detectors. In *CVPR*, 2022. 6

[36] 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 *CVPR*, 2022. 7

[37] Nico Messikommer, Daniel Gehrig, Antonio Loquercio, and Davide Scaramuzza. Event-based asynchronous sparse convolutional networks. In *ECCV*, 2020. 2

[38] Nico Messikommer, Daniel Gehrig, Mathias Gehrig, and Davide Scaramuzza. Bridging the gap between events and frames through unsupervised domain adaptation. *RA-L*, 2022. 2

[39] Ishan Misra, Abhinav Shrivastava, and Martial Hebert. Watch and learn: Semi-supervised learning for object detectors from video. In *CVPR*, 2015. 2, 14

[40] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. *IJRR*, 2017. 2

[41] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 8

[42] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1742–1750, 2015. 2

[43] Etienne Perot, Pierre De Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. Learning to detect objects with a 1 megapixel event camera. *NeurIPS*, 2020. 1, 2, 3, 5, 7, 12

[44] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds. *IEEE Journal of Solid-State Circuits*, 2010. 5

[45] Charles R Qi, Yin Zhou, Mahyar Najibi, Pei Sun, Khoa Vo, Boyang Deng, and Dragomir Anguelov. Offboard 3d object detection from point cloud sequences. In *CVPR*, 2021. 2

[46] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *CoRL*, 2018. 2

[47] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *CVPR*, 2019. 2

[48] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *TPAMI*, 2019. 2

[49] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 2015. 5

[50] Simon Schaefer, Daniel Gehrig, and Davide Scaramuzza. Aegnn: Asynchronous event-based graph neural networks. In *CVPR*, 2022. 1, 2

[51] Cedric Scheerlinck, Henri Rebecq, Daniel Gehrig, Nick Barnes, Robert Mahony, and Davide Scaramuzza. Fast image reconstruction with an event camera. In *WACV*, 2020. 2

[52] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*, 2020. 5

[53] 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. *NeurIPS*, 2015. 1, 2

[54] Krishna Kumar Singh, Fanyi Xiao, and Yong Jae Lee. Track and transfer: Watching videos to simulate strong human supervision for weakly-supervised object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3548–3556, 2016. 2

[55] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *NeurIPS*, 2020. 2

[56] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020. 2, 3, 5

[57] Timo Stoffregen, Cedric Scheerlinck, Davide Scaramuzza, Tom Drummond, Nick Barnes, Lindsay Kleeman, and Robert Mahony. Reducing the sim-to-real gap for event cameras. In *ECCV*, 2020. 2

[58] Zhaoning Sun, Nico Messikommer, Daniel Gehrig, and Davide Scaramuzza. Ess: Learning event-based semantic segmentation from still images. In *ECCV*, 2022. 2

[59] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *ECCV*, 2022. 2, 3

[60] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 2020. 14

[61] He Wang, Yezhen Cong, Or Litany, Yue Gao, and Leonidas J Guibas. 3dioumatch: Leveraging iou prediction for semi-supervised 3d object detection. In *CVPR*, 2021. 4, 5, 8

[62] Lin Wang, Yujeong Chae, Sung-Hoon Yoon, Tae-Kyun Kim, and Kuk-Jin Yoon. Evdistill: Asynchronous events to end-task learning via bidirectional reconstruction-guided cross-modal knowledge distillation. In *CVPR*, 2021. 2

[63] Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. In *CVPR*, 2023. 4, 14

[64] Wenming Weng, Yueyi Zhang, and Zhiwei Xiong. Event-based video reconstruction using transformer. In *ICCV*, 2021. 2

[65] Ziyi Wu, Xudong Liu, and Igor Gilitschenski. Eventclip: Adapting clip for event-based object recognition. *arXiv preprint arXiv:2306.06354*, 2023. 2

[66] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020. 2

[67] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. In *ICCV*, 2021. 2, 3, 4, 5, 6, 8

[68] Pengxiang Yan, Guanbin Li, Yuan Xie, Zhen Li, Chuan Wang, Tianshui Chen, and Liang Lin. Semi-supervised video salient object detection using pseudo-labels. In *ICCV*, 2019. 2, 3, 14

[69] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *TKDE*, 2022. 14

[70] Yan Yang, Liyuan Pan, and Liu Liu. Event camera data pre-training. In *ICCV*, 2023. 2

[71] Junbo Yin, Jin Fang, Dingfu Zhou, Liangjun Zhang, Cheng-Zhong Xu, Jianbing Shen, and Wenguan Wang. Semi-supervised 3d object detection with proficient teachers. In *ECCV*, 2022. 4

[72] Alessandro Zanardi, Andreas Aumiller, Julian Zilly, Andrea Censi, and Emilio Frazzoli. Cross-modal learning filters for rgb-neuromorphic wormhole learning. *RSS*, 2019. 2

[73] Jiqing Zhang, Bo Dong, Haiwei Zhang, Jianchuan Ding, Felix Heide, Baocai Yin, and Xin Yang. Spiking transformers for event-based single object tracking. In *CVPR*, 2022. 2

[74] Lunjun Zhang, Anqi Joyce Yang, Yuwen Xiong, Sergio Casas, Bin Yang, Mengye Ren, and Raquel Urtasun. Towards unsupervised object detection from lidar point clouds. In *CVPR*, 2023. 2, 12

[75] Yongqiang Zhang, Yancheng Bai, Mingli Ding, Yongqiang Li, and Bernard Ghanem. W2f: A weakly-supervised to fully-supervised framework for object detection. In *CVPR*, 2018. 2

[76] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 2018. 14

[77] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multi-vehicle stereo event camera dataset: An event camera dataset for 3d perception. *RA-L*, 2018. 8

[78] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *CVPR*, 2019. 2

[79] Alex Zihao Zhu, Ziyun Wang, Kaung Khant, and Kostas Daniilidis. Eventgan: Leveraging large scale image datasets for event cameras. In *ICCP*, 2021. 2

[80] Nikola Zubić, Daniel Gehrig, Mathias Gehrig, and Davide Scaramuzza. From chaos comes order: Ordering event representations for object recognition and detection. In *ICCV*, 2023. 1, 2, 7

A. More Implementation Details

A.1. Tracking-based Post-Processing

Given the detection outputs from TTA, we first aggregate them via Non-Maximum Suppression (NMS). Now, for each event frame I at timestep t, we have a set of 2D bounding boxes $\mathcal{B}^t = \{b_j^t = (x_j, y_j, w_j, h_j, l_j, t)\}$. We follow the tracking-by-detection paradigm [2] to build tracks by linking detection boxes between frames, which is also inspired by [74]. Each track $s_k = \{(b, v_x, v_y)_t, k, n, q\}$ maintains the following attributes: (v_x, v_y) is the estimated velocity in the pixel space, k is the track's unique ID, n is its length so far, and $q \in [0, 1]$ is its current score, which is decayed over time and determines whether to delete the track. In the first frame, we initialize each box in $\mathcal{B}^0$ as a track, where $(v_x, v_y) = (0, 0)$, $n = 1$, and $q = 0.9$. For every coming frame I_t, we need to link its bounding boxes $\mathcal{B}^t$ to existing tracks $\{s_k\}$. We first predict the new box parameter of each track using its coordinate in the last frame (x, y) and (v_x, v_y) with a linear motion assumption, while keeping its size in the last frame (w, h) unchanged. Then, we compute pairwise IoUs between the predicted boxes and $\mathcal{B}^t$ to apply greedy matching. Only boxes in the same category and with an IoU larger than τ_{iou} can be matched. For unmatched boxes, we initialize tracks for them as done in the first frame. For unmatched tracks, we decay its score as $q_t = 0.9 * q_{t-1}$, which allows for object re-identification in future frames. For matched boxes and tracks, we update the box parameters and velocity, and reset the score as $q = 1$. Finally, we go over each track and delete those with a lower score $q < \tau_{\text{del}}$. After tracking, each box is associated with a track, and thus a length n (note that n represents the number of successful matches instead of the time between creation and deletion, i.e., unmatched timesteps do not count). We identify boxes with $n < T_{trk}$ as temporally inconsistent.

Similar to TTA, we apply tracking in forward and backward event sequences, and will only remove a box if it has a short track length in both directions. For those long tracks, we inpaint boxes at their unmatched timesteps using the synthesized ones with linear motion. This builds upon the prior of object permanence and can further stabilize the training in our experiments. Overall, the detection losses related to removed and inpainted boxes will be ignored during model training. For hyper-parameters, we choose $\tau_{\text{iou}} = 0.45$ which is the same as the IoU threshold used in NMS, $\tau_{\text{del}} = 0.55$ which is slightly higher than $0.9^6 \approx 0.53$, and $T_{trk} = 6$. We do not tune these hyper-parameters and simply use the first set of values that works.

A.2. RVT Training

We build upon the open-source codebase of RVT[1] [14] and copy most of their training settings. Events in each $50ms$

time window are converted to a frame-like 10-channel event histogram representation. We use RVT-S in most of the experiments due to limited computation resources, but also scale up LEOD to RVT-B in Sec. 4.4. Following [14], we down-sample the labeling frequency of 1Mpx [43] to 10 Hz.

Pre-training on Sparse Labels. The same optimizer, batch size, data augmentation, and data sampling methods are used. In order to apply the time-flip TTA during pseudo-labeling, we add a temporal flipping augmentation. We train for 200k steps on 1% labels, 300k steps on 2% labels, and 400k steps on 5%, 10%, and 100% labels. On 1Mpx [43], we use an increased sequence length $L = 10$ for training, as we observed clearly better results compared to $L = 5$.

Pseudo Label Filtering. We filter out low-confidence bounding boxes to obtain high-quality pseudo labels. As introduced in Sec. 3.1, RVT predicts an objectness score $p_{obj} \in [0, 1]$ and a class-wise IoU score $p_{iou} \in \mathbb{R}^C, p_{iou}^i \in [0, 1]$ for each bounding box. We only keep boxes with $p_{obj} \geq \tau_{\text{hard}}$ and $\max(p_{iou}) \geq \tau_{\text{hard}}$, and further ignore losses on those with $p_{obj} < \tau_{\text{soft}}$ and $\max(p_{iou}) < \tau_{\text{soft}}$.

Self-training on Pseudo Labels. We still use the same batch size, data augmentation, and data sampling methods. Since pseudo labels have a much higher labeling frequency than the original ground-truth labels, the effective training batch size under the same event sequence length is larger. Following the square root scaling law [18], we use a higher learning rate of 5×10^{-4} on Gen1 [8] and 8×10^{-4} on 1Mpx. We train for 150k and 200k steps in round 1 and round 2 self-training, respectively. At each training step, we first conduct the normal anchor assignment process [12] to compute training losses, and then set the losses on anchors associated with uncertain boxes (boxes with a detection score lower than τ_{soft} and the ignored and inpainted boxes from tracking-based post-processing) as 0.

Training Objective of RVT. RVT adopts the anchor-free YOLOX [12] detection head. Let $o^i \in \{0, 1\}$ denote whether an anchor point is matched to a ground-truth box kept after label filtering, and $r^i \in \{0, 1\}$ denote whether it is matched to a box removed in tracking or soft anchor assignment (thus ignored in loss computation), the training loss of RVT is:

$$
\begin{aligned}
L = & \mathbb{1}_{\{r^i=0\}} L_{\text{BCE}}(p_{\text{obj}}^i, o^i) + \mathbb{1}_{\{o^i=1\}} L_{\text{CE}}(p_{\text{iou}}^i, l^i) \\
& + \mathbb{1}_{\{o^i=1\}} L_{\text{IoU}}(\Delta b^i, b^i)
\end{aligned} \tag{1}
$$

Our proposed components only bring negligible overheads to model training. Therefore, we can train our model on 2 NVIDIA A40 GPUs. The pre-training stage takes 60 hours, while self-training takes around 40 hours.

B. Detailed Analysis of Pseudo Label Quality

Fig. 9 shows the precision and recall of pseudo labels under different settings and thresholds. They are computed by
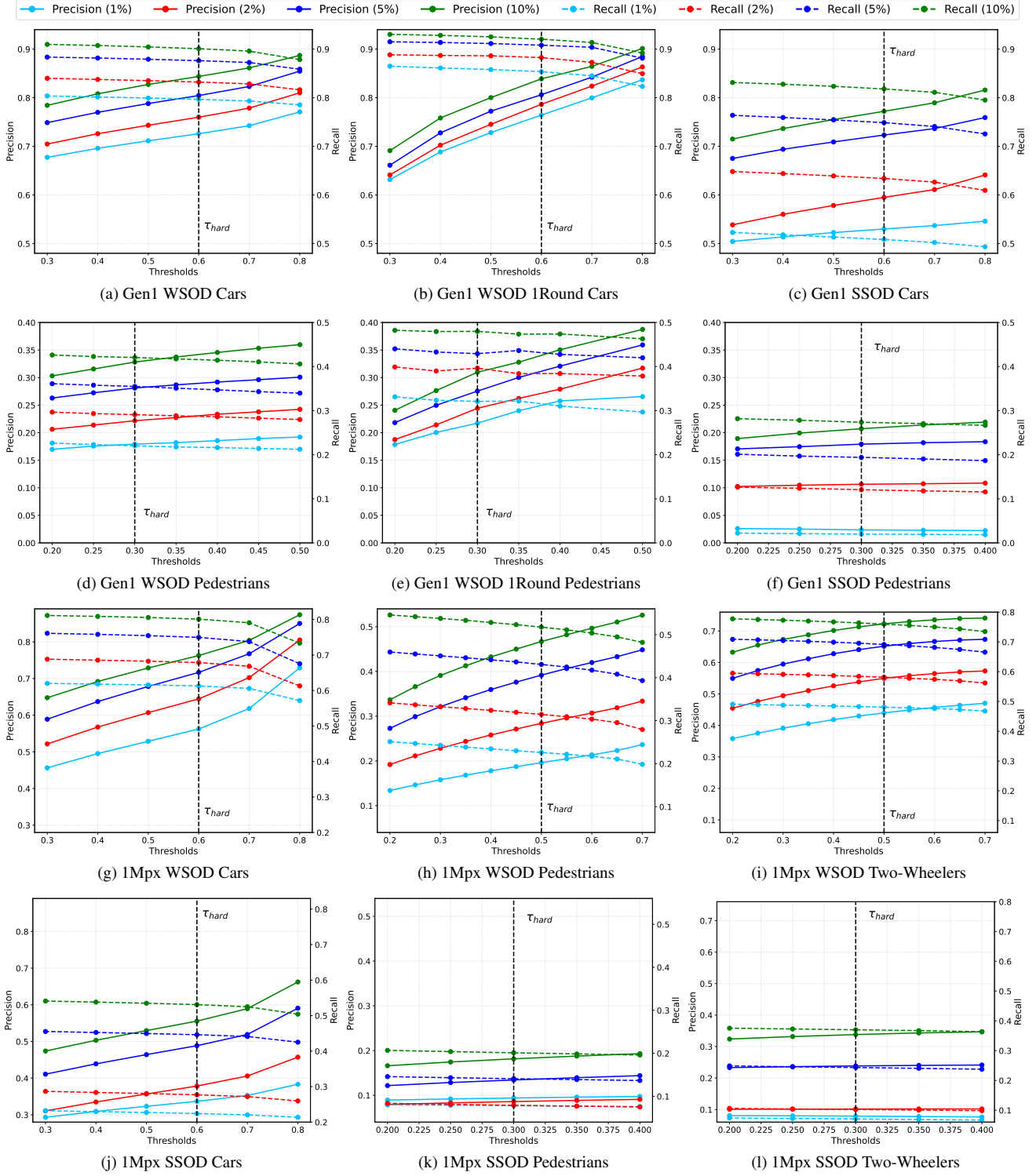
Figure 9. We plot the precision and recall of pseudo labels generated under different settings. In each figure, solid lines represent precision and dotted lines represent recall. Four labeling ratios $1\%, 2\%, 5\%, 10\%$ are selected. The black dotted line is the threshold for label filtering. We fix the Y-axis value range within each ground $\{(a), (b), (c)\}, \{(d), (e), (f)\}, \{(g), (j)\}, \{(h), (k)\}, \{(i), (l)\}$ for easy comparisons.

evaluating pseudo labels against the ground-truth labels at annotated but skipped frames. If a predicted box has an IoU higher than 0.75 with a ground-truth box, we treat it as a positive detection. We make the following observations:

More pre-training labels lead to better quality. In all settings, models pre-trained with more labels produce pseudo labels with clearly higher precision and recall.

13

Cars are much easier to detect than other categories. Comparing cars, pedestrians, and two-wheelers, it is clear that cars have a much better label quality in all settings. This is because cars are larger and there are more bounding box annotations of cars than other objects. On 1Mpx, two-wheelers are slightly easier to detect than pedestrians. Future work can study how to address the class-imbalance issue and improve detections on hard examples.

Self-training improves pseudo label quality, but may degrade precision. Comparing Fig. 9 (a) and (b), (d) and (e), we can see that one round of self-training greatly improves the recall (dotted lines). However, the precision (solid lines) drops if we use a small τ_{hard}. This is because the model learns to discover more objects after self-training, but is also over-confident in its predictions. Therefore, fewer false positives are removed in the filtering process. One solution is to increase the threshold τ_{hard} over the number of self-training rounds, as done in [63]. We tried this in our preliminary experiments but did not observe a clear improvement.

Weakly-supervised learning (WSOD) leads to better results than semi-supervised learning (SSOD). Comparing the WSOD and SSOD results in Fig. 9, we can see that models trained in WSOD produce much higher quality pseudo labels than their SSOD counterparts. Together with the detection mAP results presented in Sec. 4.2, this proves that sparsely labeling as many event streams as possible is better than densely labeling a few event sequences.

Gen1 vs. 1Mpx. Comparing Fig. 9 (a) and (g), (c) and (j), it is clear that models on Gen1 detect cars much better than on 1Mpx. This is because 1Mpx has a higher resolution and the number of cars per frame is also larger (1Mpx: 3.8 vs Gen1: 1.9). Interestingly, as can be seen from Fig. 9 (d) and (h), the label quality of pedestrians on Gen1 is worse than on 1Mpx. After visualizing some results, we realize that this is because Gen1 does not provide annotations for two-wheelers, but the model detects lots of two-wheelers as pedestrians, which are regarded as false positives. In contrast, 1Mpx does not have this issue as two-wheelers are also labeled which disambiguates model learning. Indeed, the gap in precision is much higher than recall, as precision penalizes false positives. Future work can study how to learn more discriminative features to separate object categories, e.g., with class-centric contrastive loss [33].

C. Visualization of Pseudo Labels

We visualize some pseudo labels on Gen1 in Fig. 10.

Failure case analysis. Tracking-based post-processing is able to eliminate temporally inconsistent boxes. However, since we use a fixed threshold $T_{trk} = 6$ for all tracks, some objects may be incorrectly removed. In Fig. 10 (a), the car highlighted by the purple arrow is a hard example as it only triggers a few events. The model only detects it in one frame while missing it in later frames, leading to a short

track length. As a result, the correct detection at $t = 16$ is mistakenly removed. In Fig. 10 (b), the cars coming from the other direction move very fast, and only stay visible for less than T_{trk} timesteps. Thus, they are also wrongly removed. Nevertheless, since we ignore these boxes during model training instead of suppressing them as background, such errors are less harmful. Fig. 10 (c) shows another failure case where a two-wheeler is recognized as a pedestrian as discussed in Appendix B.

Successful examples. In Fig. 10 (c), we visualize the tracking trajectory of a pedestrian (the green curve). Although the pedestrian is occluded and thus not detected at $t = 16$, our tracker is able to re-identify it at $t = 21$, thus keeping it in the pseudo labels. Fig. 10 (d) shows an example where a car is not annotated in the ground-truth labels. Our model successfully discovers it and corrects the annotation error.

D. Discussion on Experimental Setting Naming

In this paper, we propose two settings under the label-efficient event-based detection task: **(i)** weakly-supervised object detection (WSOD) where all event sequences are sparsely annotated, and **(ii)** semi-supervised object detection (SSOD) where some event sequences are densely annotated, and others are fully unlabeled. While (ii) undoubtedly belongs to semi-supervised learning, (i) may be controversial. In fact, the definition of weakly- and semi-supervised learning is often overlapping in the literature. For example, the Wikipedia page[2] seems to give similar definitions to these two tasks: "**Weak supervision**, also called **semi-supervised learning**, is a paradigm in machine learning..." Previous surveys [60, 69] identify a key property in semi-supervised learning: labeled and unlabeled data should be (although from the same distribution) independent of each other. In contrast, the labeled frames in a sparsely labeled event sequence are not independent of the unlabeled frames in the same sequence. On the other hand, another survey on weakly-supervised learning [76] regards "incomplete supervision where only a subset of training data are given with labels" as one type of weak supervision, which is similar to our sparse labeling setting. These are the main reasons we term (i) weakly-supervised learning to differentiate it from semi-supervised learning.

However, we note that some works [39, 68] learning video object detection with sparsely labeled frames call their setting semi-supervised learning. Moreover, if we employ a feedforward detector, i.e., detectors that do not leverage temporal information, setting (i) becomes closer to semi-supervised learning as labeled and unlabeled timesteps become less relevant. Nevertheless, we believe recurrent detectors are the future trend in event-based object detection as they lead to significantly stronger performance.

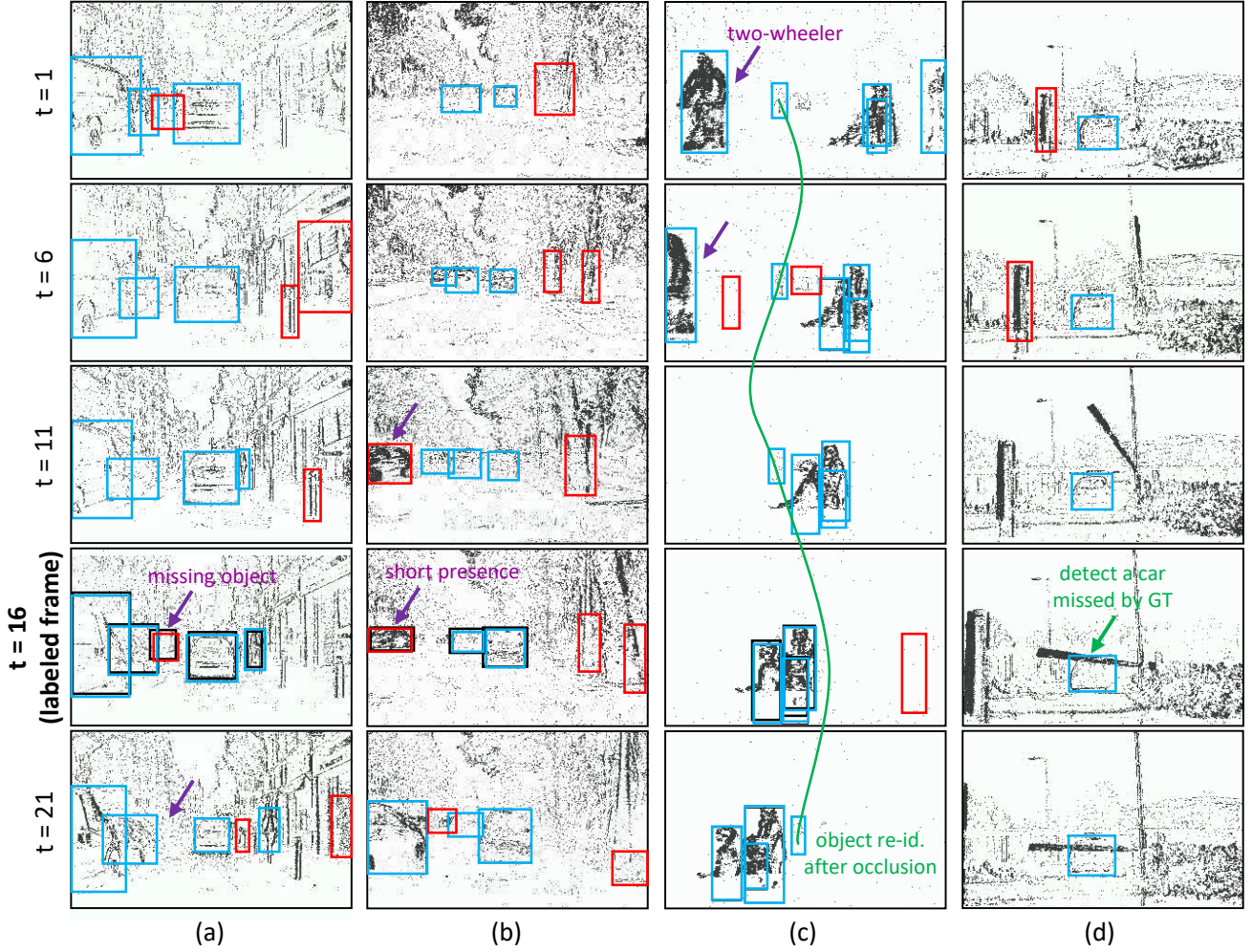[2]https://en.wikipedia.org/wiki/Weak_supervision

Figure 10. We visualize some pseudo labels on Gen1 that are generated by an RVT-S after one round of self-training. Blue boxes are pseudo labels kept for model training while red boxes are those removed by tracking-based post-processing. Black boxes at $t = 16$ are ground-truth annotations. The t here denotes timesteps of the event frame representation instead of seconds in the real world. Purple arrows highlight some failure cases of our method while green arrows highlight some desired behaviors.

E. Societal Impact

This paper proposes a framework to learn better event-based object detectors with limited labeled data. Object detection is a core task in computer vision that is used across a wide variety of applications including healthcare, entertainment, communication, mobility, and defense. While only a subset of scenarios in this application can benefit from event-camera data, it is still difficult to predict the overall impact of the technology. Moreover, event-based detectors may introduce biases that are different from those encountered in classical cameras and better understanding such biases is an open research problem. While we do not see any immediate risks of human rights or security violations introduced by our work, future work building upon it will carefully need to investigate implications on its particular application area.