GoStop: Reinforcement Learning for Adaptive Temporal Aggregation in Event-Based Feature Tracking

Youngho Kim*, Hoonhee Cho*, Jae-Young Kang, and Kuk-Jin Yoon

KAIST
{kmax2001, gnsgnsgml, kjyoon, mickeykang}@kaist.ac.kr

Abstract. Feature tracking plays a fundamental role in understanding scene motion and supports various downstream tasks. Event cameras, with their high temporal resolution and asynchronous sensing, enable low-latency and motion-robust perception, making them well-suited for feature tracking under fast and non-linear motion. However, existing event-based feature tracking methods rely on fixed heuristic rules based on hand-tuning for event accumulation. Such strategies fail to adapt to diverse motion dynamics, leading to degraded performance under abrupt motion changes or low-motion scenarios. In this paper, we model event accumulation as a sequential decision-making problem and introduce reinforcement learning (RL) framework to adaptively control the accumulation process for online event-based feature tracking. Our approach trains a RL agent that decides whether to continue accumulating events or to perform tracking inference based on motion cues. The proposed adaptive temporal agent enables dynamic adaptation to varying motion patterns without relying on hand-crafted rules. Furthermore, we introduce a Dynamic Event-based Tracking (DEFT) dataset with dynamic motion distributions to evaluate the robustness of the feature tracking. Extensive experiments demonstrate that integrating our plug-and-play framework to existing feature tracking methods consistently outperforms heuristic-based approaches, improving robustness under dynamic motion while offering a better balance between tracking accuracy and efficiency. Our project codes and datasets are available at `https://github.com/kmax2001/GoSTOP`.

Keywords: Event Camera · Reinforcement Learning · Feature Tracking

1 Introduction

Feature tracking [21, 33] estimates the trajectories of query points, enabling a deeper understanding of scene motion and supporting various downstream tasks [11,76,86]. Event cameras [10,23], with high dynamic range, low power consumption, and asynchronous per-pixel operation, enable low-latency sensing that is robust to motion blur and well-suited for fast ego and object motion [17,39,45],

* The first two authors contributed equally.

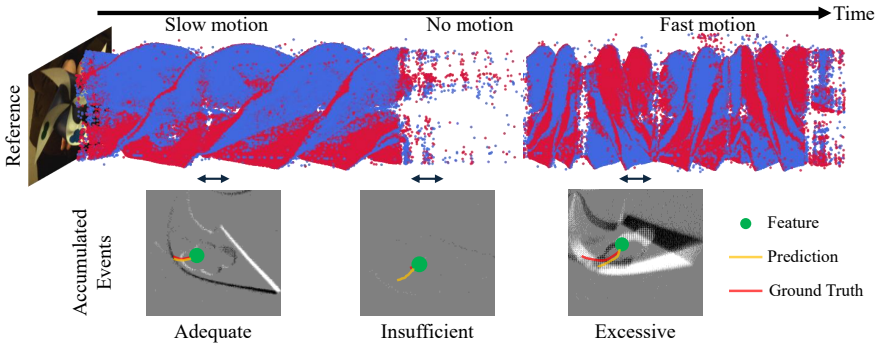

Fig. 1: Illustration of event accumulation under a fixed temporal window across different motion states. Because the same temporal window is used regardless of motion dynamics, the accumulated events can provide insufficient cues under little motion and blurred, noisy cues under fast motion. This motivates our adaptive event accumulation strategy, which adjusts the temporal window according to the observed motion.

motivating event-based feature tracking [4, 24, 25]. Recent learning-based approaches [30,31,50,64,65,82] leverage these properties to achieve accurate and efficient tracking, particularly under dynamic and non-linear motion, where frame-based cameras often struggle.

However, event cameras generate a variable amount of information depending on scene dynamics, such as object motion. This temporal variability poses a significant challenge for sequential tasks such as event-based feature tracking, where the quality of each prediction directly affects subsequent predictions. To address this issue, event-based feature tracking methods [65, 82] typically rely on manually designed rules, such as fixed-length temporal windows with hand-tuned hyperparameters. These rules are often based on prior knowledge and tuned offline, making them difficult to generalize across diverse motion patterns and dynamic scenes. As a result, they fail to adapt to varying motion dynamics, limiting the ability to fully leverage the asynchronous nature of event cameras beyond a fixed frame rate. As illustrated in Fig. 1, tracking performs well when sufficient informative events are accumulated, but degrades significantly when motion suddenly diminishes or becomes rapid. We observe that learning-based event tracking becomes unstable when motion ceases after continuous movement and also degrades under abrupt motion changes, suggesting that a fixed accumulation interval is insufficient for robust event-based tracking.

To address this limitation, we propose to adaptively determine event accumulation in event-based feature tracking by formulating it as a sequential decision-making problem, where actions at each timestep influence future states. Deciding how long to accumulate events in a causal, online tracking setting is highly challenging due to the temporal and motion dependencies of event streams and the absence of explicit supervision for optimal accumulation. To this end, we introduce reinforcement learning (RL) to event-based feature tracking for the first time. We model the accumulation process as an environment, where an agent observes the current tracking state, such as motion cues and intermediate

event statistics, and decides whether to continue accumulating events or to perform tracking inference, analogous to step-by-step decision-making. The agent is trained to optimize temporal window and maximize tracking performance through a reward function that reflects the quality of tracking outcomes, enabling the model to learn a policy that dynamically adapts accumulation strategies to diverse motion patterns while reducing reliance on hand-crafted rules and improving robustness and generalization. Compared to existing event-based feature tracking methods, our approach remains robust under varying motion distributions. By adaptively controlling the accumulation process, it gathers sufficient information for reliable inference while avoiding unnecessary accumulation, leading to a favorable trade-off between accuracy and efficiency. Furthermore, our method can be readily integrated into existing well-designed event-based tracking frameworks, making it a general and extensible solution that can be applied to future event-based feature tracking systems. The core contributions of our work can be summarized as follows:

- We propose a RL-based event-based feature tracking framework that adaptively determines the event accumulation process in a causal and online manner.
- We introduce a challenging dynamic event-based tracking (DEFT) dataset with diverse and rapidly changing motion distributions, including complex and non-linear motion patterns.

2 Related Works

Event-based Feature Tracking. Frame-based feature tracking [6,21,33,41,42] is inherently limited by the camera frame rate, making it difficult to handle non-linear motion. In contrast, event cameras provide high temporal resolution, enabling event-based tracking methods to operate at high effective frame rates and better capture complex motion. Early event-based approaches [3, 18, 19, 36, 80, 90] relied on hand-crafted methods, which require extensive parameter tuning and often generalize poorly to new environments. Recent learning-based methods have reduced the need for such tuning. For instance, Deep-EvTracker [65] demonstrates strong cross-dataset generalization using data-driven learning. ETAP [30] leverages temporal correlations across event sequences to achieve high accuracy, but incurs high computational cost, limiting the benefits of high temporal resolution of event cameras. More recent work, BlinkTrack [82], improves robustness to occlusions and efficiency through better training data and filtering strategies.

Despite these advances, existing methods [7, 31, 51, 54, 59, 87, 89, 101] rely on predefined criteria to determine the accumulation duration, which limits their ability to adapt to varying motion dynamics. In this work, we address this limitation by introducing a reinforcement learning-based approach that adaptively determines the temporal window for event accumulation, enabling robust tracking across diverse motion patterns. Furthermore, existing evaluation datasets [35,66] often exhibit relatively uniform motion patterns, limiting their ability to assess

performance under diverse and dynamic conditions. To address this, we introduce a dynamic event-based tracking (DEFT) dataset, which captures realistic scenarios with diverse ever-changing motion distributions within each sequence.

Temporal Windowing of Event Streams. Processing event streams typically requires aggregating asynchronous events into temporal windows to construct dense event representations [22, 60, 61, 94, 106, 109] for downstream tasks such as object detection [2, 14, 26, 29, 40, 56, 69, 93, 108], recognition [16, 43, 44, 72, 103, 104], depth estimation [5, 13, 15, 27, 52, 67, 95, 96, 107] and scene reconstruction [32, 37, 53, 97, 99]. A common strategy is to accumulate events over a fixed temporal interval or according to predefined criteria, enabling the use of conventional vision pipelines while simplifying the inherently continuous nature of event data.

However, the role of temporal windowing differs significantly across tasks. For tasks such as detection, predictions are made based on information within a given temporal window, and the choice of the window boundaries is relatively flexible. In contrast, feature tracking inherently estimates the motion of points over time, where each prediction corresponds to the displacement between consecutive states. As a result, the starting point of each temporal window is naturally defined by the previous estimation, making the choice of the accumulation interval fundamentally more constrained. This property makes tracking particularly sensitive to the temporal extent of event accumulation. A short temporal window may fail to capture sufficient motion information, leading to unstable estimates, while a long temporal window can blur motion dynamics or degrade performance under non-linear motion. Therefore, under varying motion dynamics, fixed temporal windowing strategies are insufficient for robust event-based feature tracking.

Reinforcement Learning for Vision Tasks. Reinforcement learning (RL) has been widely adopted in computer vision [8, 34, 81, 83] for problems where explicit supervision is unavailable or difficult to define. In tasks such as object detection [20, 62, 68, 71, 75], visual tracking [58, 63, 100], and active perception [12, 28, 47, 81, 91, 92], the optimal action cannot be directly specified, and its quality is evaluated only through its impact on the final performance. As a result, models must learn appropriate actions without ground-truth labels for each decision, relying instead on feedback signals to optimize long-term objectives. RL provides a principled framework for addressing such settings by enabling policy learning from reward signals, and has been used to learn behaviors such as selecting informative regions [9, 63], controlling viewpoints [81], and state control [49, 70, 102], where the correctness of each action is inherently ambiguous.

This challenge naturally arises in event-based feature tracking. Given a continuous and asynchronous event stream, determining how much temporal information to accumulate for reliable inference does not have a well-defined ground-truth, and the quality of a given temporal window can only be assessed through the resulting tracking performance. However, existing methods rely on fixed heuristics for temporal aggregation, which limits their ability to adapt to varying motion dynamics. In this work, we address this limitation by leveraging RL to adaptively determine the temporal window for event accumulation. This en-

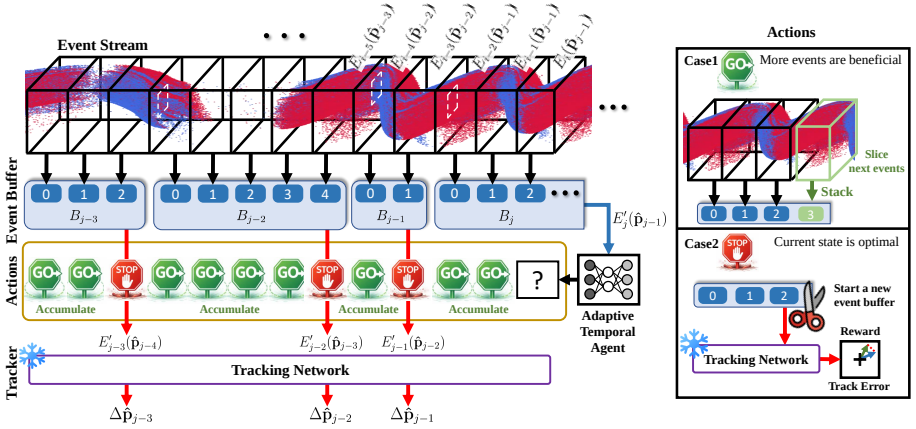

Fig. 2: Overview of the proposed framework. Events are accumulated in a buffer, and the Adaptive Temporal Agent decides whether to trigger tracking inference. It performs inference when sufficient information is available, delays updates otherwise, and increases inference frequency under rapid motion for stable tracking.

ables the model to learn when to perform inference without explicit supervision, leading to improved robustness under diverse motion conditions.

3 Methodology

Despite extensive research on event-based feature tracking, maintaining stable and high tracking performance in real-world scenarios remains challenging due to motion-induced variability in triggered events and limited computational resources, particularly when balancing performance with computational efficiency. To address this, we propose a RL–based approach in which a deep neural agent exploits the temporal dynamics of event data and adaptively accumulates events for robust tracking. The agent optimizes the tracker input online, effectively resolving the trade-off between inference time and tracking performance. The remainder of this section is organized as follows. In Sec. 3.1, we present the underlying tracking module with an adaptive temporal window, and in Sec. 3.2, we describe the reinforcement learning framework for optimizing temporal aggregation of events.

3.1 Event-based Feature Tracking with Adaptive Temporal Window

Event-based Feature Tracking. Conventional event-based feature tracking aims to estimate the motion of visual features within small patches. At current timestamp $t = t_i$, tracker receives event input $E_i(\hat{\mathbf{p}}_{i-1})$, generated within the temporal window $[t_{i-1}, t_i]$ of fixed length inside a square patch centered at $\hat{\mathbf{p}}_{i-1} = (x_{i-1}, y_{i-1})$. The model predicts the feature displacement $\Delta\hat{\mathbf{p}}_i = (\Delta\hat{x}_i, \Delta\hat{y}_i)$ and updates the patch center as $\hat{\mathbf{p}}_i = \hat{\mathbf{p}}_{i-1} + \Delta\hat{\mathbf{p}}_i$, which is then used to obtain the next input $E_{i+1}(\hat{\mathbf{p}}_i)$. Because each prediction determines the input

for subsequent steps, error between the predicted center and the ground-truth center accumulates over time. To improve the robustness of event-based feature tracking, we propose a method that controls the temporal window at the input level of the event module. Conventional tracking approaches [65, 82] use event data $E_i(\hat{\mathbf{p}}_{i-1})$ from a fixed temporal interval $T_i = [t_i - \Delta t, t_i]$ to predict motion at each timestamp t_i. In contrast, we decouple inference from fixed timestamps by introducing an inference index j, where each index corresponds to a tracker inference step. At each step j, the agent maintains a buffer B_j that accumulates incoming event streams until sufficient motion information is observed. Instead of performing inference at every timestamp, the agent continuously aggregates events into the current buffer and evaluates motion cues from the buffered events. The agent triggers tracker inference only when the buffer contains sufficient information; once inference is triggered, the buffer is flushed and a new buffer B_{j+1} is initialized for subsequent event accumulation. At each inference step j, the tracker receives the event stream $E'_j(\hat{\mathbf{p}}_{j-1})$ within the optimized temporal window $[t_{j-1}, t_j]$ and outputs the displacement $\Delta\hat{\mathbf{p}}_j$ to update the patch center $\hat{\mathbf{p}}_j$. The temporal window is defined by the previous inference timestamp, $t_{j-1} = t_j - \Delta t \cdot l_j$, where l_j denotes the number of event accumulations stored in B_j. To enable fine-grained control of the temporal window, we set Δt to be half of that used in conventional trackers. By learning adaptive temporal windows, the agent dynamically adjusts the amount of event information according to scene motion, reducing overall inference time while improving tracking accuracy. This simple control mechanism enables asynchronous operation of the tracking module, consistent with the event camera's sensing paradigm based on intensity changes.

Sequential Decision Problem Formulation. For effective temporal window control, we introduce a reinforcement learning (RL) agent that decides whether to perform inference (*stop*) using a pretrained event-based tracker or accumulate the next event stream (*go*). The RL agent is designed to encode the tendency of triggered events and determine whether the accumulated event stream is sufficient for reliable motion estimation under varying event densities. We formulate temporal window control as a sequential decision-making problem, where the agent selects an action $a \in A$ at state $s \in S$. The state s represents the events accumulated in the buffer B_j, and the action space A consists of two discrete actions: *go* and *stop*. Each action is evaluated using a reward function $r(s, a)$, and future rewards are discounted by a factor γ. Our agent is optimized using two networks, a policy network and a critic network. The policy network parameterizes the policy $\pi(a|s)$, which represents the probability distribution over actions given the current state. The critic network approximates the value function $V^\pi(s_i)$, which estimates the expected sum over the discounted rewards from the state s_i over trajectories $\tau(\pi)$ induced by policy π.

$$V^\pi(s_i) = \mathbb{E}_{\tau(\pi)}[\sum_{t=i}^{\infty} \gamma^t r(s_t, a_t)]. \tag{1}$$

3.2 Reinforcement Learning for Adaptive Temporal Agent

State Design. Many prior works [55, 57, 79, 98] employ feature embeddings extracted from pretrained networks as state representations for reinforcement learning agents. A straightforward approach in our setting would be to leverage features from a pretrained tracking encoder as the state representation. However, such intermediate representations introduce several limitations. First, features extracted from the tracking network are optimized for the tracking objective, which may not align with the decision-making process of the RL agent, thereby introducing irrelevant or misleading information. Second, computing such features with complex encoders incurs additional computational overhead and latency, which is undesirable for high-rate event streams and can hinder timely decision making.

For effective decision-making under a limited computational budget, we design a lightweight architecture which enables fast decision-making and adopt the event representation [82] with the current accumulation information, providing the policy network with rich motion cues. We encode the 2D event representation $E'_j(\hat{\mathbf{p}}_{j-1}) \in \mathbb{R}^{P\times P\times C}$ using 2D convolution layers, where C denotes the number of event channels and P denotes the patch size. The resulting feature vector is concatenated with the temporal window information and fed into a multi-layer perceptron (MLP) to produce the final action. This lightweight policy architecture reduces the overall inference time despite the additional computation introduced by the decision making.

Reward Design. The desirable behavior is accumulating event data until sufficient motion information is accumulated while avoiding excessive accumulation that may degrade tracking reliability. The agent must balance *go* and *stop* actions to trade off between frequent inference for accuracy and infrequent inference for efficiency. To achieve this, we design a simple and intuitive reward function:

$$r = \begin{cases} L_j \cdot \max\left(-1, 0.6 - e_i\right) - \lambda & \text{if } a_i = stop \\ 0 & \text{if } a_i = go \end{cases}, \quad e_i = \frac{1}{P}||\hat{\mathbf{p}}_i - \mathbf{p}_i||_2 \tag{2}$$

where e_i denotes the tracking error at current timestamp t_i and L_j denotes the information of event stacks in B_j. Incorporating this error into the reward encourages decisions that improve tracking performance. This formulation encourages the agent to continue accumulating events as long as the tracking module can reliably estimate motion. When the accumulated events are insufficient and noisy, the agent selects *go* which increases L_j. Conversely, when the temporal window becomes excessively large, both L_j and e_i tend to increase, resulting in a large negative reward in the next *stop*. However, this formulation may bias the policy toward excessive use of the *stop* action. To prevent this behavior, we introduce an additional penalty term λ, which discourages overuse of *stop* which can lead to unreliable estimation.

Policy Optimization. We train the RL agent using the on-policy algorithm Proximal Policy Optimization (PPO) [78]. PPO is a widely adopted policy gradient algorithm known for its training stability, which is achieved through con-

servative policy updates by clipping policy ratio. The policy function is primarily optimized by maximizing the following clipped surrogate objective:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t], \\ r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}. \tag{3}$$

The advantage function $\hat{A}_t$ measures how beneficial action a_t is at state s_t. It is estimated using Generalized Advantage Estimation (GAE) [77] whose computation requires value predictions from the critic network. The advantage function evaluates action quality considering the difficulty of the state itself [84]. When the advantage is positive, the gradient flows in the way that increases the probability $\pi_\theta(a = a_t|s = s_t)$. Meanwhile, a negative advantage reduces the probability of unreliable actions, leading to stable and efficient policy improvement.

Privileged Information for Critic. During the policy optimization, tracking failure may occur for two different reasons. In some cases, the failure is caused by an inappropriate temporal window due to the control failure of the RL agent. In other cases, however, the agent chooses a reasonable temporal window while the tracker still drifts to an incorrect feature due to factors such as occlusion or interference from nearby structures. In latter cases, the agent receives a large negative reward even though the failure is not caused by its decision. Since the advantage estimate in Eq. (3) is computed from the value function, such noisy rewards can introduce inconsistent value targets, which in turn destabilize the policy learning process. Although one could attempt to design a reward function considering these edge cases, such manual reward shaping may instead bias the agent toward unintended suboptimal behaviors.

For stable value function approximation, we provide the critic network with additional privileged information available *only during training* to help distinguish the causes of tracking failure and take into account the inherent difficulty of the scene. Specifically, we provide the critic with ground-truth motion information for the feature tracking task, along with the remaining timesteps in the sequence. The ground-truth motion information includes the motion speed of the target feature, occlusion information and the tracking error, allowing the critic to infer the difficulty of the scene, such as rapid motion and occlusion. The remaining timesteps indicate how much time remains in the sequence, enabling the critic to better assess the long-term impact of the current action. Importantly, this privileged information is used only during training and is accessible solely to the critic, but not to the policy. With this privileged information, well-fitted critic network effectively guides the policy network toward the desired behavior.

4 Dynamic Event-based Feature Tracking(DEFT) dataset

Although real datasets [35,66] have been proposed for event-based feature tracking, the motion patterns in these datasets are often relatively simple and lack the diversity commonly observed in real-world scenarios. We observe that conventional tracking methods frequently fail when deployed in real environments

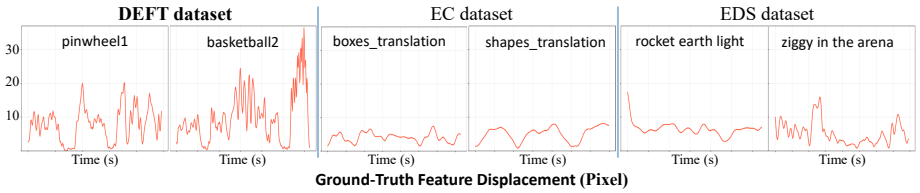

Fig. 3: Average feature displacement over time for selected scenes, illustrating representative motion distributions. EDS is slightly more dynamic than EC, but both are largely uniform, whereas DEFT exhibits more diverse and abrupt motion patterns, highlighting the strengths of event cameras.

with highly variable feature motion. To address this limitation, we introduce a new real-world event-based feature tracking dataset, the Dynamic Event-based Tracking dataset (DEFT), which also serves as a comprehensive benchmark for evaluation. DEFT is captured using a DAVIS346C camera, which provides 346 × 260 asynchronous event streams. The dataset contains approximately 57 seconds of event stream data across eight scenarios which have diverse object motion. Ground-truth feature trajectories are manually annotated at 25 Hz. As shown in Fig. 3, trajectories in DEFT exhibit significantly diverse motion patterns including abrupt stops, rapid accelerations, and frequent motion changes that produce highly dynamic event streams. In addition, the dataset contains heterogeneous motion from both foreground objects and background elements, making the tracking task substantially more challenging and establishing it as a strong benchmark for evaluating real-world event-based applications. More details on the DEFT dataset are provided in the supplementary material.

5 Experiments

5.1 Implementation Details

We train our agent using the on-policy algorithm Proximal Policy Optimization (PPO) [78], implemented in Stable-Baselines3 [73]. Training is performed on a single NVIDIA TITAN RTX 24GB GPU with 16 parallel environments and takes approximately 12 hours. For training, an event-stream sequence is clipped to 95 event frames. The agent is trained for 5×10^6 iterations where one iteration denotes one state transition. We use the Adam optimizer [46] with a learning rate linearly decayed from 1e-4 to 1e-5, and set the discount factor $\gamma = 0.9$ so that our agent considers future rewards as well. The policy network and critic network consist of two CNN encoders with ReLU activations [1], which encode the event representation into a 2D feature vector, followed by three MLP layers that output the action and value function, respectively. We freeze the parameter of a pre-trained tracker during RL training. For fair comparison with conventional event-based tracking methods, we adopt event representation consistently with previous works [65,82,88]. We set $\Delta t = 0.005s$ for high temporal resolution of the asynchronous tracking. In Eq. (2), we set $\lambda = 0.2$ for reward function and $L_j = 2 \cdot l_j$.

Table 1: Performance comparison on the DEFT dataset. We report Feature Age (FA) and Expected Feature Age (EFA). All methods use only event data. DT and BT denote Deep-EV-Tracker [65] and BlinkTrack [82], respectively, and K.F denotes the Kalman Filter. Bold and underlined values indicate the best and second-best results, respectively. Values in parentheses indicate improvements over the base model.

Method	Desk		Pinwheel1		Pinwheel2		Pinwheel3		Basketball1		Basketball2		RobotArm1		RobotArm2		Average	
	FA↑	EFA↑	FA↑	EFA↑	FA↑	EFA↑	FA↑	EFA↑	FA↑	EFA↑	FA↑	EFA↑	FA↑	EFA↑	FA↑	EFA↑	FA↑	EFA↑
HASTE [4]	0.565	0.562	0.021	0.020	0.244	0.243	0.379	0.379	0.196	0.196	0.034	0.034	0.079	0.079	<u>0.435</u>	<u>0.435</u>	0.244	0.243
DT [65]	0.172	0.170	0.016	0.016	0.131	0.131	0.135	0.134	0.052	0.052	0.024	0.024	0.061	0.060	0.092	0.091	0.086 (-)	0.085 (-)
DT+**Ours**	<u>0.590</u>	<u>0.584</u>	<u>0.298</u>	0.279	0.391	<u>0.387</u>	<u>0.398</u>	<u>0.398</u>	<u>0.697</u>	<u>0.696</u>	0.370	0.370	<u>0.572</u>	<u>0.572</u>	0.399	0.396	<u>0.465</u> (+0.379)	<u>0.460</u> (+0.375)
BT [82]	0.157	0.157	0.293	<u>0.293</u>	0.398	0.382	0.188	0.186	0.644	0.639	0.488	0.478	0.557	0.541	0.300	0.300	0.378 (-)	0.372 (-)
BT+K.F [82]	0.157	0.157	0.245	0.245	<u>0.400</u>	0.384	0.182	0.181	0.669	0.664	<u>0.510</u>	<u>0.500</u>	0.558	0.543	0.295	0.294	0.377 (−0.001)	0.371 (−0.001)
BT+**Ours**	**0.836**	**0.835**	**0.405**	**0.405**	**0.639**	**0.635**	**0.460**	**0.459**	**0.809**	**0.808**	**0.638**	**0.634**	**0.739**	**0.738**	**0.608**	**0.608**	**0.642** (+0.264)	**0.640** (+0.268)

5.2 Experimental Setup

Baselines. We evaluate our method in a plug-and-play manner on top of recent event-based feature trackings, including BlinkTrack [82] and its concurrent work Deep-EV-Tracker [65]. By integrating our RL-based temporal window control module into these learnable trackers, we assess the performance gains achieved without modifying their underlying architectures. We additionally compare against BlinkTrack augmented with a Kalman filter [38], which has been shown to improve tracking performance on the EC and EDS datasets. Furthermore, we include learning-free baseline, HASTE [4], which represent rule-based event feature tracking approaches.

Datasets. Prior event-based feature trackings typically adopt a cross-domain generalization setting, where training and evaluation datasets are separated. In this work, we follow the same cross-domain setting and use the MultiTrack dataset [82] for training, as in prior work, to demonstrate the effectiveness of our method without introducing additional training data. The tracker is adopted using pretrained models provided by the original authors, whose parameters are kept fixed during training, while the training dataset is used solely to learn the decision-making policy of the RL agent. We evaluate our method on three real-world event-based tracking datasets: DEFT, EC [66], and EDS [35].

Metric. We follow the evaluation protocol of previous event-based feature trackings [65, 82]. Since the EC, EDS, and DEFT datasets provide ground-truth annotations at a lower temporal resolution than the event-based predictions, we interpolate the predicted trajectories to the timestamps of the ground-truth annotations and compute the tracking error in pixel space. We report two standard metrics: Feature Age (FA) and Expected Feature Age (EFA). Feature Age measures the proportion of time during which the tracking error remains below a predefined threshold [65]. Expected Feature Age is defined as the product of feature age and the ratio of stable tracks whose distance remains below the threshold across all predictions.

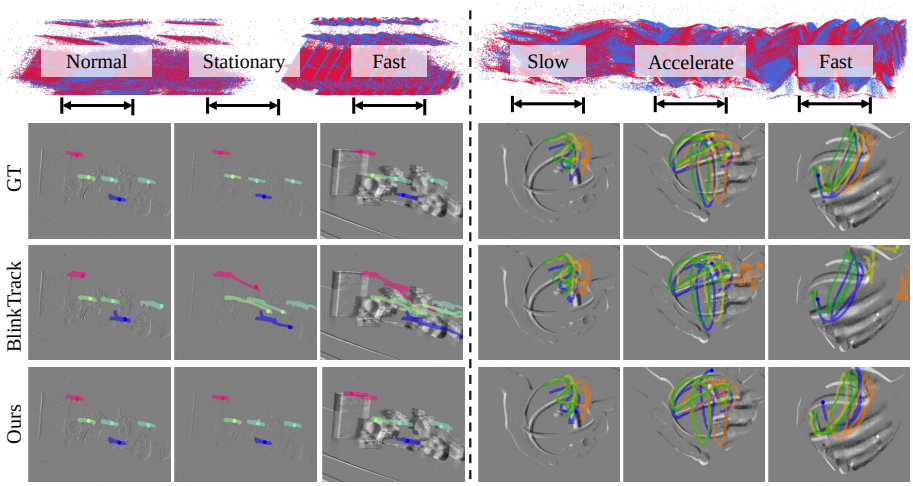

Fig. 4: Qualitative comparison on the DEFT dataset. For clarity, feature tracking results are visualized over the corresponding interval, and only the most recent events within the current duration are shown.

5.3 Experimental Results

Results on DEFT. Table 1 presents tracking performance under varying motion distributions, covering a wide range of motion dynamics. Both Deep-EV-Tracker (DT) [65] and BlinkTrack (BT) [82] struggle under challenging conditions, and in certain scenes, their performance degrades significantly, sometimes failing to produce reliable tracking results. When motion changes abruptly within a sequence, the tracker often loses reliable features, resulting in low average performance with FA of 0.086 and EFA of 0.085. Incorporating the proposed adaptive temporal agent improves robustness, allowing the model to handle challenging sequences, increasing FA and EFA by 0.379 and 0.375 on average. BT is relatively more robust than DT and performs reasonably well across several sequences. However, it still struggles under highly dynamic motion and shows limited overall performance. Even when combined with a Kalman Filter [38,82], the model cannot effectively handle such motion patterns. In contrast, integrating the proposed method with BT consistently improves tracking in all sequences, increasing FA and EFA by 0.264 and 0.268. These results indicate that the proposed RL-based temporal controller adaptively shapes the raw event stream into informative temporal segments based on the distribution of triggered events, resulting in more reliable and accurate feature tracking. We visualize the qualitative comparisons in Fig. 4. BT struggles in real-world scenarios with highly variable motion. In particular, it often fails when a moving object suddenly stops or abruptly accelerates, resulting in sparse or ambiguous event inputs. In contrast, our method adaptively adjusts the temporal window, enabling robust tracking under abrupt motion changes as well as both sparse and highly dense event conditions.

Table 2: Performance on the conventional EDS and EC benchmarks. Higher is better.

Method	EDS		EC		Average	
	FA↑	EFA↑	FA↑	EFA↑	FA↑	EFA↑
ICP [48]	0.060	0.040	0.256	0.245	0.158	0.143
EM-ICP [105]	0.161	0.120	0.337	0.334	0.249	0.227
HASTE [4]	0.096	0.063	0.442	0.427	0.269	0.245
DT [65]	0.549	0.451	0.795	0.787	0.672 (-)	0.619 (-)
DT [65]+**Ours**	<u>0.622</u>	<u>0.513</u>	**0.864**	**0.858**	**0.743** (+0.071)	**0.686** (+0.067)
BT [82]	0.568	0.474	0.833	0.819	0.701 (-)	0.647 (-)
BT [82]+K.F	0.569	0.475	<u>0.835</u>	<u>0.820</u>	0.702 (+0.001)	0.648 (+0.001)
BT [82]+**Ours**	**0.649**	**0.534**	0.819	0.811	<u>0.734</u> (+0.033)	<u>0.672</u> (+0.025)

Results on EC and EDS. Table 2 compares our method with existing approaches on the EC and EDS, which mainly contain relatively monotonic motion patterns. On these benchmarks, ours improves the average performance of both trackers. With the proposed method, DT achieves the highest score among all methods on the EC dataset, demonstrating a substantial performance gain. This result suggests that the proposed adaptive aggregation improves performance not only in challenging scenarios but also in relatively monotone scenarios.

5.4 Ablation Study and Analysis

We evaluate the effectiveness of the proposed method through experiments using BlinkTrack on the DEFT dataset, along with ablation studies and analyses.

State Design Analysis. Table 3 presents an analysis of the state representations used as input to the policy network. We consider three types of state representations: event representation, features extracted from the tracker encoder and the output of the correlation layer in the tracker. As shown in the results, using event representation yields the best performance. We attribute this to the fact that features extracted by the tracking encoder are optimized for the tracking objective, which may not be well aligned with the decision-making process of the RL agent. In contrast, event representation preserves more direct and informative cues for policy learning. As a result, the policy network can effectively learn the decision-making process without relying on additional encoders, enabling a lightweight and efficient framework.

Action Penalty Analysis. We conduct experiments on the action penalty coefficient λ in Eq. (2), which discourages excessive use of the *stop* action. As shown in Table 4, without the penalty term, the agent tends to overuse the *stop* action, leading to suboptimal tracking performance. Introducing the penalty encourages a better balance between *go* and *stop*, allowing the tracker to make predictions only after sufficient event information has been accumulated. In particular, setting $\lambda = 0.2$ achieves a favorable trade-off between *go* and *stop* actions.

Comparison between RL Agent and Diverse Time-Interval Training. One possible way to improve robustness to dynamic and diverse motion distributions is to expose the model to a wide range of temporal intervals during training. To examine this, we extend the original training setting by dynamically sampling temporal intervals within each sequence, ranging from $\times 0.5$ to $\times 2.5$ the base interval. We then compare this strategy with the proposed RL agent.

Table 3: Analysis of the effect of state representation on performance.

State Representation	FA↑	EFA↑
Event Representation	**0.642**	**0.640**
Encoder Features	0.308	0.306
Correlation	0.171	0.169

Table 4: Analysis of action penalty, λ in Eq. (2).

Action Penalty	FA↑	EFA↑
0	0.307	0.295
0.1	0.600	0.595
0.2	**0.642**	**0.640**
0.3	0.614	0.611

Table 5: Comparison of our method with data augmentation methods.

Setting	FA↑	EFA↑
BlinkTrack [82]	0.378	0.372
BlinkTrack w/ Aug.	0.191	0.189
BlinkTrack + **Ours**	**0.642**	**0.640**

Table 6: Ablation study on privileged information for the critic network.

Privileged Information		FA↑	EFA↑
GT Motion Information	Remaining Timestep		
		0.456	0.455
	✓	0.491	0.490
✓		0.572	0.567
✓	✓	**0.642**	**0.640**

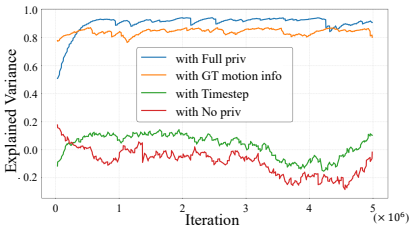

Fig. 5: Training curves of the explained variance with and without privileged information.

As shown in Table 5, such time-interval augmentation does not effectively resolve the observed failure cases. In fact, training with a broad distribution of temporal intervals generally degrades overall performance. This suggests that simply increasing temporal diversity during training is insufficient, highlighting the advantage of the proposed adaptive temporal agent.

Privileged Information for the Critic Network. We conduct an ablation study on the privileged inputs provided to the critic network. Specifically, we progressively remove each type of privileged information from the critic input and report the resulting performance degradation in Table 6. The results show that each privileged component contributes to the overall performance gain. As shown in Fig. 5, privileged information helps the critic network better approximate the expected return, as indicated by the higher explained variance. As discussed in prior work [74, 85], this demonstrates that a more accurate value function provides stronger guidance for the policy network to select actions that maximize the expected return.

Runtime Analysis. Our RL agent requires approximately 0.4 ms (2,500 FPS) per preprocessed event patch to decide *go* or *stop* on a single NVIDIA TITAN RTX. Under the same hardware and experimental settings, the tracker module [82] requires 5.3 ms (188 FPS). This shows that the proposed policy network introduces only negligible inference overhead. Owing to its lightweight design, the policy network effectively reduces unnecessary computations in the tracking module.

Adaptive Temporal Policy vs. Search-based Static Strategies. In general, prior studies [65, 82] adopt a fixed temporal interval and perform inference at regular time steps for simplicity and computational efficiency. As shown in Table 7, we extend these methods to various temporal intervals and also consider number-based event stacking [88]. For time-based stacking, performance varies

Table 7: Comparison of the trade-off between performance and runtime across various rule-based slicing strategies and our method on the DEFT dataset. Runtime is measured as the inference time required to process a single event-patch subsequence of 1 second. For time-based slicing, inference is performed at fixed temporal intervals, resulting in identical runtime regardless of the input event distribution. For number-based event slicing, runtime varies significantly depending on the number of events in each segment. To account for this variability, we report runtimes measured on segments whose event counts fall within the top 10% and top 50% of the entire dataset. † denotes the base setting of BlinkTrack [82].

Metrics	Number (Count)					Time (Second)					**Ours**
	100	250	500	1000	2500	0.001	0.005	0.01†	0.05	0.1	Adaptive
FA↑	0.542	0.580	0.585	0.461	0.207	0.100	0.121	0.378	0.384	0.209	0.642
EFA↑	0.540	0.577	0.582	0.458	0.205	0.043	0.108	0.372	0.383	0.209	0.640
Runtime (s) ↓ [Top 10% Events]	18.111	7.084	3.611	1.780	0.360	5.436	1.046	0.528	0.107	0.058	0.812
Runtime (s) ↓ [Top 50% Events]	3.080	1.196	0.631	0.318	0.150						0.516

significantly with the chosen interval, indicating that tracking accuracy is highly sensitive to this parameter and that selecting an optimal value is non-trivial. For number-based stacking, inference is triggered once a fixed number of events is accumulated. While this can improve robustness, it leads to excessive updates under high event density, causing a substantial increase in runtime and making it impractical for real-time applications. In particular, processing events corresponding to 1 second may take over 3–7 seconds, making it difficult to operate as an online feature tracking system. Such delays continuously accumulate over time, further degrading practical usability. In contrast to the base setting (†) with a fixed 0.01-second interval, our method adaptively shortens the inference interval under rapid motion, enabling successful tracking in scenarios where the base setting often fails. Although this increases the runtime due to more frequent updates, our method remains within the 1-second event segment duration, making it practical for online operation. Furthermore, when the event density is low, the inference frequency is reduced, resulting in even lower runtime than the base setting.

6 Conclusion

In this paper, we present a RL-based event feature tracking framework that adapts to the varying amount of event information encountered in real-world scenarios. With the adaptive temporal agent that dynamically controls the temporal window, the tracking module achieves both improved accuracy and reduced inference time across the real event dataset. Our learning-based adaptive temporal window consistently outperforms the hand-crafted rule-based alternatives in both practicality and efficiency. In addition, we present DEFT, our novel event-based feature tracking dataset designed to capture variable motion patterns in real-world scenarios. We hope that our RL-based temporal control framework encourages broader research of event-based robust perception in real-world applications.

7 Acknowledgments.

This work was supported by the Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2024-00457882, AI Research Hub Project), by the InnoCORE program of the Ministry of Science and ICT(N10250156), and by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2026-25473963).

References

1. Agarap, A.F.: Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375 (2018)
2. Ahmed, S.H., Finkbeiner, J., Neftci, E.: Efficient event-based object detection: a hybrid neural network with spatial and temporal attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13970–13979 (2025)
3. Alzugaray, I., Chli, M.: Ace: An efficient asynchronous corner tracker for event cameras. In: 2018 International Conference on 3D Vision (3DV). pp. 653–661. IEEE (2018)
4. Alzugaray, I., Chli, M.: Haste: multi-hypothesis asynchronous speeded-up tracking of events. In: 31st British Machine Vision Virtual Conference (BMVC 2020). p. 744. ETH Zurich, Institute of Robotics and Intelligent Systems (2020)
5. Bartolomei, L., Mannocci, E., Tosi, F., Poggi, M., Mattoccia, S.: Depth anyevent: A cross-modal distillation paradigm for event-based monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19669–19678 (2025)
6. Bian, W., Huang, Z., Shi, X., Dong, Y., Li, Y., Li, H.: Context-pips: persistent independent particles demands spatial context features. Advances in Neural Information Processing Systems **36**, 55285–55298 (2023)
7. Burner, L., Mitrokhin, A., Fermüller, C., Aloimonos, Y.: Evimo2: an event camera dataset for motion segmentation, optical flow, structure from motion, and visual inertial odometry in indoor scenes with monocular or stereo algorithms. arXiv preprint arXiv:2205.03467 (2022)
8. Caicedo, J.C., Lazebnik, S.: Active object localization with deep reinforcement learning. In: Proceedings of the IEEE international conference on computer vision. pp. 2488–2496 (2015)
9. Caicedo, J.C., Lazebnik, S.: Active object localization with deep reinforcement learning. In: Proceedings of the IEEE international conference on computer vision. pp. 2488–2496 (2015)
10. Chakravarthi, B., Verma, A.A., Daniilidis, K., Fermuller, C., Yang, Y.: Recent event camera innovations: A survey. In: European conference on computer vision. pp. 342–376. Springer (2024)
11. Chen, W., Chen, L., Wang, R., Pollefeys, M.: Leap-vo: Long-term effective any point tracking for visual odometry. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19844–19853 (2024)
12. Cheng, R., Agarwal, A., Fragkiadaki, K.: Reinforcement learning of active vision for manipulating objects under occlusions. In: Conference on robot learning. pp. 422–431. PMLR (2018)

13. Cho, H., Cho, J., Yoon, K.J.: Learning adaptive dense event stereo from the image domain. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17797–17807 (2023)
14. Cho, H., Kang, J.y., Kim, Y., Yoon, K.J.: Ev-3dod: Pushing the temporal boundaries of 3d object detection with event cameras. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 27197–27210 (2025)
15. Cho, H., Kang, J.Y., Yoon, K.J.: Temporal event stereo via joint learning with stereoscopic flow. In: European Conference on Computer Vision. pp. 294–314. Springer (2024)
16. Cho, H., Kim, H., Chae, Y., Yoon, K.J.: Label-free event-based object recognition via joint learning with image reconstruction from events. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19866–19877 (2023)
17. Cho, H., Kim, T., Jeong, Y., Yoon, K.J.: A benchmark dataset for event-guided human pose estimation and tracking in extreme conditions. Advances in Neural Information Processing Systems **37**, 134826–134840 (2024)
18. Chui, J., Klenk, S., Cremers, D.: Event-based feature tracking in continuous time with sliding window optimization. arXiv preprint arXiv:2107.04536 (2021)
19. Dardelet, L., Benosman, R., Ieng, S.H.: An event-by-event feature detection and tracking invariant to motion direction and velocity. Authorea Preprints (2021)
20. Ding, W., Majcherczyk, N., Deshpande, M., Qi, X., Zhao, D., Madhivanan, R., Sen, A.: Learning to view: Decision transformers for active object detection. arXiv preprint arXiv:2301.09544 (2023)
21. Doersch, C., Gupta, A., Markeeva, L., Recasens, A., Smaira, L., Aytar, Y., Carreira, J., Zisserman, A., Yang, Y.: Tap-vid: A benchmark for tracking any point in a video. Advances in Neural Information Processing Systems **35**, 13610–13626 (2022)
22. Fan, R., Hao, W., Guan, J., Rui, L., Gu, L., Wu, T., Zeng, F., Zhu, Z.: Eventpillars: Pillar-based efficient representations for event data. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 2861–2869 (2025)
23. Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A.J., Conradt, J., Daniilidis, K., et al.: Event-based vision: A survey. IEEE transactions on pattern analysis and machine intelligence **44**(1), 154–180 (2020)
24. Gehrig, D., Rebecq, H., Gallego, G., Scaramuzza, D.: Asynchronous, photometric feature tracking using events and frames. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 750–765 (2018)
25. Gehrig, D., Rebecq, H., Gallego, G., Scaramuzza, D.: Eklt: Asynchronous photometric feature tracking using events and frames. International Journal of Computer Vision **128**(3), 601–618 (2020)
26. Gehrig, M., Scaramuzza, D.: Recurrent vision transformers for object detection with event cameras. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13884–13893 (2023)
27. Ghosh, S., Gallego, G.: Event-based stereo depth estimation: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025)
28. Grimes, M.K., Modayil, J.V., Mirowski, P.W., Rao, D., Hadsell, R.: Learning to look by self-prediction. Transactions on Machine Learning Research (2023)
29. Hamaguchi, R., Furukawa, Y., Onishi, M., Sakurada, K.: Hierarchical neural memory network for low latency event processing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22867–22876 (2023)

30. Hamann, F., Gehrig, D., Febryanto, F., Daniilidis, K., Gallego, G.: Etap: Event-based tracking of any point. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 27186–27196 (2025)
31. Han, H., Zhai, W., Cao, Y., Li, B., Zha, Z.j.: Mate: Motion-augmented temporal consistency for event-based point tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8340–8349 (2025)
32. Han, H., Li, J., Wei, H., Ji, X.: Event-3dgs: Event-based 3d reconstruction using 3d gaussian splatting. Advances in Neural Information Processing Systems **37**, 128139–128159 (2024)
33. Harley, A.W., Fang, Z., Fragkiadaki, K.: Particle video revisited: Tracking through occlusions using point trajectories. In: European Conference on Computer Vision. pp. 59–75. Springer (2022)
34. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
35. Hidalgo-Carrió, J., Gallego, G., Scaramuzza, D.: Event-aided direct sparse odometry. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5781–5790 (2022)
36. Hu, S., Kim, Y., Lim, H., Lee, A.J., Myung, H.: ecdt: Event clustering for simultaneous feature detection and tracking. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 3808–3815. IEEE (2022)
37. Huang, J., Dong, C., Chen, X., Liu, P.: Inceventgs: Pose-free gaussian splatting from a single event camera. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 26933–26942 (2025)
38. Kalman, R.E.: A new approach to linear filtering and prediction problems (1960)
39. Kang, J.Y., Cho, H., Lee, T., Kang, M., Wen, B., Kim, Y., Yoon, K.J.: Event6d: Event-based novel object 6d pose tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15091–15104 (2026)
40. Kang, J.Y., Cho, H., Yoon, K.J.: Unleashing the temporal potential of stereo event cameras for continuous-time 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6869–6881 (2025)
41. Karaev, N., Makarov, Y., Wang, J., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6013–6022 (2025)
42. Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker: It is better to track together. In: European conference on computer vision. pp. 18–35. Springer (2024)
43. Kim, J., Bae, J., Park, G., Zhang, D., Kim, Y.M.: N-imagenet: Towards robust, fine-grained object recognition with event cameras. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2146–2156 (2021)
44. Kim, J., Hwang, I., Kim, Y.M.: Ev-tta: Test-time adaptation for event-based object recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17745–17754 (2022)
45. Kim, Y., Cho, H., Yoon, K.J.: From sharp to blur: Unsupervised domain adaptation for 2d human pose estimation under extreme motion blur using event cameras. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9406–9417 (2025)
46. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

47. Krueger, D., Leike, J., Evans, O., Salvatier, J.: Active reinforcement learning: Observing rewards at a cost (2020), https://arxiv.org/abs/2011.06709
48. Kueng, B., Mueggler, E., Gallego, G., Scaramuzza, D.: Low-latency visual odometry using event-based feature tracks. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 16–23. IEEE (2016)
49. Lee, K., Shin, U., Lee, B.U.: Learning to control camera exposure via reinforcement learning. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2975–2983 (2024), https://api.semanticscholar.org/CorpusID:268856547
50. Li, S., Zhou, Z., Xue, Z., Li, Y., Du, S., Gao, Y.: 3d feature tracking via event camera. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18974–18983 (2024)
51. Li, W., Liao, B., Zhou, Y., Wan, P., Liu, P.: E-moflow: Learning egomotion and optical flow from event data via implicit regularization. Advances in Neural Information Processing Systems **38**, 30350–30376 (2026)
52. Liang, J., Yu, B., Yang, S., Zhuang, H., Ren, J., Duan, P., Shi, B.: Eventups: Uncalibrated photometric stereo using an event camera. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7516–7525 (2025)
53. Liao, B., Zhai, W., Wan, Z., Cheng, Z., Yang, W., Zhang, T., Cao, Y., Zha, Z.J.: Ef-3dgs: Event-aided free-trajectory 3d gaussian splatting. arXiv preprint arXiv:2410.15392 (2024)
54. Liu, H., Xu, J., Chang, Y., Zhou, H., Zhao, H., Wang, L., Yan, L.: Timetracker: Event-based continuous point tracking for video frame interpolation with nonlinear motion. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 17649–17659 (2025)
55. Liu, M., Chen, Z., Cheng, X., Ji, Y., Qiu, R., Yang, R., Wang, X.: Visual whole-body control for legged loco-manipulation. The 8th Conference on Robot Learning (2024)
56. Lu, D., Kong, L., Lee, G.H., Simon Chane, C., Ooi, W.: Flexevent: towards flexible event-frame object detection at varying operational frequencies. Advances in Neural Information Processing Systems **38**, 89009–89044 (2026)
57. Luo, H., Zhou, B., Lu, Z.: Pre-trained visual dynamics representations for efficient policy learning. In: European Conference on Computer Vision. pp. 249–267. Springer (2024)
58. Luo, W., Sun, P., Zhong, F., Liu, W., Zhang, T., Wang, Y.: End-to-end active object tracking via reinforcement learning. In: International conference on machine learning. pp. 3286–3295. PMLR (2018)
59. Luo, X., Luo, A., Wang, Z., Lin, C., Zeng, B., Liu, S.: Efficient meshflow and optical flow estimation from event cameras. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19198–19207 (2024)
60. Manderscheid, J., Sironi, A., Bourdis, N., Migliore, D., Lepetit, V.: Speed invariant time surface for learning to detect corner points with event-based cameras. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10245–10254 (2019)
61. Maqueda, A.I., Loquercio, A., Gallego, G., García, N., Scaramuzza, D.: Event-based vision meets deep learning on steering prediction for self-driving cars. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5419–5427 (2018)
62. Mathe, S., Pirinen, A., Sminchisescu, C.: Reinforcement learning for visual object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2894–2902 (2016)

63. Messikommer*, N., Cioffi*, G., Gehrig, M., Scaramuzza, D.: Reinforcement learning meets visual odometry. European Conference on Computer Vision. (ECCV) (2024)
64. Messikommer, N., Fang, C., Gehrig, M., Cioffi, G., Scaramuzza, D.: Data-driven feature tracking for event cameras with and without frames. IEEE Transactions on Pattern Analysis and Machine Intelligence **47**(5), 3706–3717 (2025)
65. Messikommer, N., Fang, C., Gehrig, M., Scaramuzza, D.: Data-driven feature tracking for event cameras. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5642–5651 (2023)
66. Mueggler, E., Rebecq, H., Gallego, G., Delbruck, T., Scaramuzza, D.: The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. The International journal of robotics research **36**(2), 142–149 (2017)
67. Nam, Y., Mostafavi, M., Yoon, K.J., Choi, J.: Stereo depth from events cameras: Concentrate and focus on the future. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6114–6123 (2022)
68. Novkovic, T., Pautrat, R., Furrer, F., Breyer, M., Siegwart, R., Nieto, J.: Object finding in cluttered scenes using interactive perception. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 8338–8344. IEEE (2020)
69. Peng, Y., Li, H., Zhang, Y., Sun, X., Wu, F.: Scene adaptive sparse transformer for event-based object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16794–16804 (2024)
70. Pinto, L., Andrychowicz, M., Welinder, P., Zaremba, W., Abbeel, P.: Asymmetric actor critic for image-based robot learning. arXiv preprint arXiv:1710.06542 (2017)
71. Pitcher, T., Förster, J., Chung, J.J.: Reinforcement learning for active search and grasp in clutter. In: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 7301–7306. IEEE (2024)
72. Plizzari, C., Planamente, M., Goletto, G., Cannici, M., Gusso, E., Matteucci, M., Caputo, B.: E2 (go) motion: Motion augmented event stream for egocentric action recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19935–19947 (2022)
73. Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., Dormann, N.: Stable-baselines3: Reliable reinforcement learning implementations. Journal of machine learning research **22**(268), 1–8 (2021)
74. Salter, S., Rao, D., Wulfmeier, M., Hadsell, R., Posner, I.: Attention-privileged reinforcement learning. In: Conference on Robot Learning. pp. 394–408. PMLR (2021)
75. Samiei, M., Li, R.: Object detection with deep reinforcement learning. arXiv preprint arXiv:2208.04511 (2022)
76. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: European conference on computer vision. pp. 501–518. Springer (2016)
77. Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P.: High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438 (2015)
78. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)

79. Seo, Y., Lee, K., James, S.L., Abbeel, P.: Reinforcement learning with action-free pre-training from videos. In: International Conference on Machine Learning. pp. 19561–19579. PMLR (2022)
80. Seok, H., Lim, J.: Robust feature tracking in dvs event stream using bézier mapping. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1658–1667 (2020)
81. Shang, J., Ryoo, M.S.: Active vision reinforcement learning under limited visual observability. Advances in Neural Information Processing Systems **36**, 10316–10338 (2023)
82. Shen, Y., Li, Y., Chen, S., Li, G., Huang, Z., Bao, H., Cui, Z., Zhang, G.: Blinktrack: Feature tracking over 80 fps via events and images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9298–9308 (2025)
83. Song, G., Myeong, H., Lee, K.M.: Seednet: Automatic seed generation with deep reinforcement learning for robust interactive segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1760–1768 (2018)
84. Sutton, R.S., McAllester, D., Singh, S., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. Advances in neural information processing systems **12** (1999)
85. Vapnik, V., Vashist, A.: A new learning paradigm: Learning using privileged information. Neural networks **22**(5-6), 544–557 (2009)
86. Vecerik, M., Doersch, C., Yang, Y., Davchev, T., Aytar, Y., Zhou, G., Hadsell, R., Agapito, L., Scholz, J.: Robotap: Tracking arbitrary points for few-shot visual imitation. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 5397–5403. IEEE (2024)
87. Wan, Z., Luo, J., Dai, Y., Lee, G.H.: Event-aided dense and continuous point tracking: Everywhere and anytime. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7936–7946 (2025)
88. Wang, L., Ho, Y.S., Yoon, K.J., et al.: Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10081–10090 (2019)
89. Wang, X., Wang, S., Tang, C., Zhu, L., Jiang, B., Tian, Y., Tang, J.: Event stream-based visual object tracking: A high-resolution benchmark dataset and a novel baseline. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19248–19257 (2024)
90. Wang, Z., Molloy, T., van Goor, P., Mahony, R.: Event blob tracking: An asynchronous real-time algorithm. arXiv preprint arXiv:2307.10593 (2023)
91. Wang, Z., Hamann, F., Chaney, K., Jiang, W., Gallego, G., Daniilidis, K.: Event-based continuous color video decompression from single frames. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 4968–4978 (2025)
92. Whitehead, S.D., Ballard, D.H.: Active perception and reinforcement learning. In: Machine learning proceedings 1990, pp. 179–188. Elsevier (1990)
93. Wu, Z., Gehrig, M., Lyu, Q., Liu, X., Gilitschenski, I.: Leod: Label-efficient object detection for event cameras. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16933–16943 (2024)
94. Xu, N., Wang, L., Yao, Z., Okatani, T.: Mets: Motion-encoded time-surface for event-based high-speed pose tracking. International Journal of Computer Vision **133**(7), 4401–4419 (2025)
95. Yan, Z., Jiao, J., Wang, Z., Lee, G.H.: Event-driven dynamic scene depth completion. arXiv preprint arXiv:2505.13279 (2025)

96. Yu, B., Ren, J., Han, J., Wang, F., Liang, J., Shi, B.: Eventps: Real-time photometric stereo using an event camera. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9602–9611 (2024)
97. Yu, W., Feng, C., Li, J., Tang, J., Yang, J., Tang, Z., Cao, M., Jia, X., Yang, Y., Yuan, L., et al.: Evagaussians: Event stream assisted gaussian splatting from blurry images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 24780–24790 (2025)
98. Yuan, Z., Xue, Z., Yuan, B., Wang, X., Wu, Y., Gao, Y., Xu, H.: Pre-trained image encoder for generalizable visual reinforcement learning. Advances in Neural Information Processing Systems **35**, 13022–13037 (2022)
99. Yura, T., Mirzaei, A., Gilitschenski, I.: Eventsplat: 3d gaussian splatting from moving event cameras for real-time rendering. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26876–26886 (2025)
100. Zhang, D., Maei, H., Wang, X., Wang, Y.F.: Deep reinforcement learning for visual object tracking in videos. arXiv preprint arXiv:1701.08936 (2017)
101. Zhang, J., Wang, Y., Liu, W., Li, M., Bai, J., Yin, B., Yang, X.: Frame-event alignment and fusion network for high frame rate tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9781–9790 (2023)
102. Zhang, S., Meng, Q., Liu, Q., Nie, L., Zhong, B., Fan, X., Ji, R.: Interactive object placement with reinforcement learning. In: International Conference on Machine Learning (2023), https://api.semanticscholar.org/CorpusID:260842090
103. Zheng, X., Wang, L.: Eventdance: Unsupervised source-free cross-modal adaptation for event-based object recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17448–17458 (2024)
104. Zhou, J., Zheng, X., Lyu, Y., Wang, L.: Exact: Language-guided conceptual reasoning and uncertainty estimation for event-based action recognition and more. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18633–18643 (2024)
105. Zhu, A.Z., Atanasov, N., Daniilidis, K.: Event-based feature tracking with probabilistic data association. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 4465–4470. IEEE (2017)
106. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Unsupervised event-based learning of optical flow, depth, and egomotion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 989–997 (2019)
107. Zhu, J., Pan, T., Cao, Z., Liu, Y., Kwok, J.T., Xiong, H.: Depth any event stream: Enhancing event-based monocular depth estimation via dense-to-sparse distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5146–5155 (2025)
108. Zhu, L., Wang, X., Wang, L., Huang, H., et al.: Rethinking scale-aware temporal encoding for event-based object detection. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems
109. Zubić, N., Gehrig, D., Gehrig, M., Scaramuzza, D.: From chaos comes order: Ordering event representations for object recognition and detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12846–12856 (2023)