

FACET

Fast and Accurate Event-Based Eye Tracking Using Ellipse Modeling for Extended Reality

Ding, Junyuan; Wang, Ziteng; Gao, Chang; Liu, Min; Chen, Qinyu

DOI

[10.1109/ICRA55743.2025.11127327](https://doi.org/10.1109/ICRA55743.2025.11127327)

Publication date

2025

Document Version

Final published version

Published in

Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA)

Citation (APA)

Ding, J., Wang, Z., Gao, C., Liu, M., & Chen, Q. (2025). FACET: Fast and Accurate Event-Based Eye Tracking Using Ellipse Modeling for Extended Reality. In *Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 10347-10354). (Proceedings - IEEE International Conference on Robotics and Automation). IEEE. <https://doi.org/10.1109/ICRA55743.2025.11127327>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

**Green Open Access added to [TU Delft Institutional Repository](#)
as part of the Taverne amendment.**

More information about this copyright law amendment
can be found at <https://www.openaccess.nl>.

Otherwise as indicated in the copyright section:
the publisher is the copyright holder of this work and the
author uses the Dutch legislation to make this work public.

FACET: Fast and Accurate Event-Based Eye Tracking Using Ellipse Modeling for Extended Reality

Junyuan Ding¹, Ziteng Wang², Chang Gao³, Min Liu², Qinyu Chen^{4*}

Abstract—Eye tracking is a key technology for gaze-based interactions in Extended Reality (XR), but traditional frame-based systems struggle to meet XR’s demands for high accuracy, low latency, and power efficiency. Event cameras offer a promising alternative due to their high temporal resolution and low power consumption. In this paper, we present FACET (Fast and Accurate Event-based Eye Tracking), an end-to-end neural network that directly outputs pupil ellipse parameters from event data, optimized for real-time XR applications. The ellipse output can be directly used in subsequent ellipse-based pupil trackers. We enhance the EV-Eye dataset by expanding annotated data and converting original mask labels to ellipse-based annotations to train the model. Besides, a novel trigonometric loss is adopted to address angle discontinuities and a fast causal event volume event representation method is put forward. On the enhanced EV-Eye test set, FACET achieves an average pupil center error of 0.20 pixels and an inference time of 0.53 ms, reducing pixel error and inference time by 1.6× and 1.8× compared to the prior art, EV-Eye, with 4.4× and 11.7× less parameters and arithmetic operations. The code is available at <https://github.com/DeanJY/FACET>.

I. INTRODUCTION

Recently, Extended Reality (XR) is rapidly transforming the way people perceive and interact with the digital world. Eye tracking, a technology that measures and records eye movements, has become indispensable for immersive XR experiences [1], [2], especially after the introduction of the Apple Vision Pro in June 2023 [3]. Eye tracking enables gaze-based interactions in XR environments [4]–[6], allowing users to control and navigate virtual spaces simply by directing their gaze. While efforts continue to integrate this technology into wearable devices [7]–[9], challenges like latency and power consumption must be addressed to ensure smooth and effective experience.

The human eye is the fastest-moving organ, capable of movements exceeding 300°/s [10]. Capturing these rapid movements accurately requires a frame rate of kilo-hertz to ensure smooth tracking and reduce motion sickness in virtual environments [11]. However, achieving such a high frame rate is challenging for wearable devices, which must operate at low power levels, typically in the milliwatt range.

*Qinyu Chen is the corresponding author.

¹Junyuan Ding is with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China dingjunyuan@buaa.edu.cn

²Ziteng Wang and Min Liu are with DVSense (Beijing) Technology Co., Ltd., China.

³Chang Gao is with the Department of Microelectronics, Delft University of Technology, The Netherlands.

⁴Qinyu Chen is with the Leiden Institute of Advanced Computer Science (LIACS), Leiden University, The Netherlands. q.chen@liacs.leidenuniv.nl

Most head-mounted devices (HMDs) rely on frame-based eye-tracking systems. A recent study reports tracking delays between 45 and 81 ms in various HMD eye trackers [12], which falls short of the kilo-hertz frame rate needed for accurate eye movement capture. Additionally, frame-based sensors capable of reaching kilo-hertz consume substantial power. The large data volumes also require high bandwidth and significant energy for transfer and processing, posing challenges for real-time applications on wearable devices.

Event cameras [13], also known as Dynamic Vision Sensors (DVS), offer an effective and efficient alternative for solving eye-tracking challenges. By capturing only brightness changes, they generate sparse asynchronous events, providing high temporal resolution and low power consumption. These unique characteristics make event cameras highly suitable for high-speed, low-power eye tracking: they produce less data and reduce processing needs during fixation while still capturing fast and subtle eye movements during saccades. Previous event-based eye-tracking studies have shown promising results [14]–[24]. However, most of them use detection neural networks to detect the pupil in every step. The high computational cost of neural networks prevents these models from achieving higher frequency. [20], [21] use simple ellipse-based trackers to track the pupil for most steps and employ neural network inference only when the pupil tracking is lost, for example after a blink. This detection-tracking schema significantly reduces the computational burden. However, these two methods use a segmentation model to acquire the mask of the pupil and then fit its ellipse boundary. Compared to lightweight detection models like MobileNet series [25]–[27], segmentation models (e.g. U-Net [28]) have larger computational cost. It also does not take advantage of the fact that event cameras emphasize the boundaries of pupils.

To fully take advantage of the event data, we propose FACET, Fast and Accurate Event-based Eye Tracking, a lightweight pupil detector that takes in events and outputs ellipse prediction of pupils, which is not only lighter and faster, but also can be trained end-to-end. This detector can be directly fitted into the existing detection-tracking eye-tracking schema. The main contributions of this work are as follows:

- We introduced a fast and accurate end-to-end event-based pupil detector using ellipse modeling.
- We proposed a dataset enhancement method that uses a semi-supervised approach to expand the annotated data and convert the mask labels to ellipse-based annotations. We used this method to label all 1.5 million samples in

the EV-Eye dataset, while the original dataset only has over 9,000 labeled samples.

- We proposed trigonometric loss to address the discontinuity problem in angle prediction for ellipse parameters.
- We designed a fast causal event volume method for event accumulation to regularize the distribution of event representation values.

II. RELATED WORKS

A. Frame-based Eye Tracking Method

Traditional frame-based eye tracking typically utilizes frame-based cameras to capture eye movements, with two common approaches: model-based and appearance-based eye tracking. Model-based eye tracking [29]–[31] locates key points corresponding to the eye’s geometrical features and fits them to an eye model using optimization techniques. These methods have limitations in headsets, which often require manual calibration and struggle with variations in eye shape and lighting conditions. The appearance-based method [32]–[36] focuses on the visual appearance of the eye, with a trend of using deep learning techniques to track the eye within the raw image. It requires substantial training data and the model can be computationally intensive and leads to large processing latency. Additionally, these frame-based methods often require high-resolution cameras, which can be both expensive and cumbersome for mobile devices. Moreover, the frame rate of the standard frame-based camera generally peaks at 200 Hz. Cameras with higher frame rates consume significant power, often at watt levels, which exceeds the milli-watt power budget for a mobile eye tracking system.

B. Event-based Eye Tracking Method

Event-based eye tracking utilizes the sparse data stream from DVS for high frame rates with much less bandwidth, offering greater energy efficiency than traditional frame-based systems. 3ET [14] introduces a sparse change-based convolutional Long-Short-Term-Memory (LSTM) model for event-based eye tracking, which reduces arithmetic operations by approximately 4.7 \times , compared to a standard convolutional LSTM, without losing accuracy, however, uses synthetic event data and fixed time windows, limiting its ability to meet kilo-hertz frame rate demands. Retina [15] introduces a neuromorphic approach that integrates a directly trained Spiking Neuron Network (SNN) regression model and leverages a state-of-the-art low power edge neuromorphic processor, achieving 5 mW power consumption and around 3-pixel error on the INI-30 dataset with 64 \times 64 resolution. Lightweight models [19], [22] propose spatio-temporal convolution and bidirectional selective recurrent models, respectively, both with approximately 3-pixel error with 60 \times 80 resolution of the 3ET+ dataset [18]. [23] tracks the eye movements by detecting and tracking the corneal glint; however, it requires illumination from a flashing light source.

On the other hand, frameworks [17], [20] have combined both frame and event data for eye tracking, utilizing geometric fitting techniques and segmentation networks,

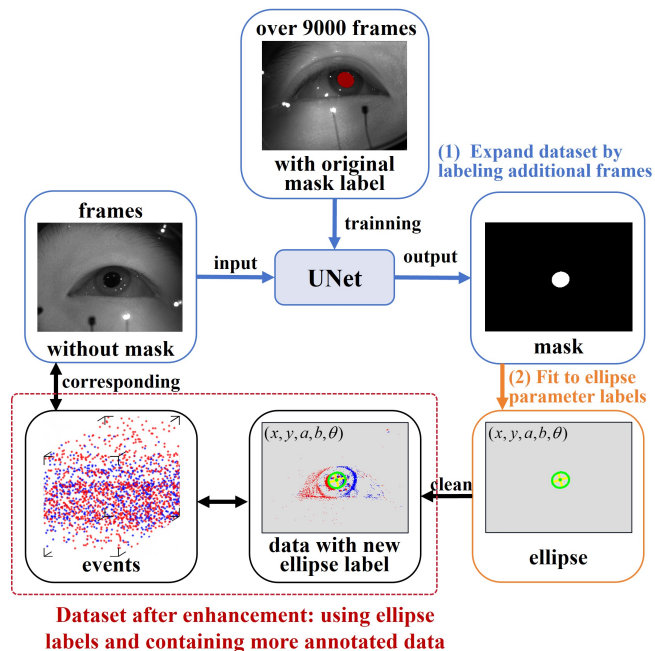


Fig. 1. Flowchart for expanding the EV-Eye dataset and annotating it with ellipse labels. We first trained a U-Net segmentation network using over 9,000 frames with mask labels, enabling it to generate masks for other unlabeled frames. Then, we fitted these masks into ellipses to obtain five parameters (x, y, a, b, θ) . Finally, we annotated the events corresponding to these frames with the ellipse labels generated by the U-Net to produce more annotated event data.

respectively. These approaches demonstrate the advantages of combining event data with frame data, achieving both high frame rates and satisfactory accuracy but struggle with addressing the issue of power consumption.

III. DATASET

EV-Eye [20], the largest existing event-based eye-tracking dataset, contains data from 48 individuals with a diverse range of genders and ages. The dataset includes over 1.5 million near-eye grayscale images and 2.7 billion event samples captured using two DAVIS346 event cameras. Frames are timestamped at 40 ms intervals, synchronized with the corresponding event data. Although the EV-Eye dataset [20] provides a valuable foundation, it has limitations that hinder its direct application to our project. It contains only around 9,000 frames with annotated pupil segmentation, which is insufficient for training robust models. The labels are full-size segmentation masks, while subsequent tracking modules require ellipse predictions as input.

To address these issues, we improved the dataset in two key ways: (1) we expanded it by labeling additional frames, and (2) we converted the full-image pupil segmentation masks into ellipse parameter labels. Fig. 1 illustrates the process of generating the updated dataset. A semi-supervised learning approach was employed to utilize the large volume of unlabeled data effectively. Pupil segmentations on unlabeled grayscale images were obtained using a U-Net model [28] trained on the labeled frames in this dataset. All

pupil segmentation labels are fitted to ellipses expressed in (x, y, a, b, θ) format, where (x, y) represent the ellipse's center coordinates, a and b are the lengths of the major and minor axes ($a \geq b$), and $\theta \in [0^\circ, 180^\circ)$ denotes the rotation angle. Inaccurate labels are manually removed. From the updated EV-Eye dataset, 20,000 samples are randomly selected for the training set, 5,000 for the validation set, and 5,000 for the test set.

IV. METHOD

This section covers the processing of events in Section IV-A, the network architecture in Section IV-B, and the loss function in Section IV-C. An overview of the entire framework is shown in Fig. 2.

A. Event Processing

1) *Event Binning Method*: To prepare event data for neural network input, events are divided into bins and accumulated into representations. Choosing the appropriate binning duration is important: Short bins may lack sufficient data, while long bins can reduce frame rate and introduce excessive noise. Previous works [14], [22] use fixed time interval binning to maintain consistent frame rates. However, this approach has limitations: when there is no eye movement, no events are generated, yet the model still consumes resources on unnecessary inference; during eye movements, a large volume of events is produced in a short time, increasing computational load. In our FACET framework, we utilize a fixed-count binning method. This allows the model to avoid wasting resources on unnecessary inference when no events are generated due to the lack of eye movement.

2) *Event Accumulation Method*: One method to accumulate events in bins into representations is the event volume [38]. However, the event volume at time t takes events both before and after t , which is impossible during real time processing. Due to the temporal causality of event sequences, causal event volume [19] using only events before t is more suitable for real-time processing. Building upon this approach, FACET proposes a fast causal event volume that further reduces the time required for event accumulation.

For an event bin B containing n events $E = \{e_i | i = 1 \dots n\}$ in a period of Δt , where $E_i = (x_i, y_i, p_i, t_i)$ represents the coordinates, polarity, and timestamp of the number i event in the bin, causal event volume accumulates all the events in the bin to a 2D representation. $p_i = 0$ means a negative event showing the pixel gets dimmer at that time, while $p_i = 1$ means a positive event showing the pixel gets brighter. For the pixel at (x, y) , the value of the causal event volume at the end of the event bin t is calculated as follows:

$$V_{pos}(x, y) = \sum_{\{E_i | p_i=1\}} \delta_{x_i, x} \cdot \delta_{y_i, y} \cdot k\left(\frac{t-t_i}{\Delta t}\right) \quad (1)$$

$$V_{neg}(x, y) = \sum_{\{E_i | p_i=0\}} \delta_{x_i, x} \cdot \delta_{y_i, y} \cdot k\left(\frac{t-t_i}{\Delta t}\right) \quad (2)$$

δ represents the Kronecker delta function, in which $\delta_{ij} = 1$ only if $i = j$, otherwise $\delta_{ij} = 0$. The kernel function $k(\tau)$ and

Algorithm 1 Fast Causal Event Volume

Input: $E = \{e_i | i = 1 \dots n\}$ ($e_i = (x_i, y_i, p_i, t_i)$)
 l : limit
Param: c : contribution
Output: V_{pos}, V_{neg}

```

1: for  $e_i$  in  $E$  do
2:   if  $p_i$  is positive then
3:     if  $V_{pos}[x_i, y_i] + c_i \leq l$  then
4:        $V_{pos}[x_i, y_i] \leftarrow V_{pos}[x_i, y_i] + c_i$ 
5:     end if
6:   else
7:     if  $V_{neg}[x_i, y_i] + c_i \leq l$  then
8:        $V_{neg}[x_i, y_i] \leftarrow V_{neg}[x_i, y_i] + c_i$ 
9:     end if
10:  end if
11: end for
12: return  $V_{pos}, V_{neg}$ 

```

Heaviside step function $H(x)$ are defined as:

$$k(\tau) = H(\tau) \max(1 - |\tau|, 0) \quad (3)$$

$$H(x) = \begin{cases} 0, & \text{if } x < 0 \\ 1, & \text{if } x \geq 0 \end{cases} \quad (4)$$

here, V_{pos} and V_{neg} represent the accumulated contribution of positive and negative polarity events, respectively, to the occupancy of the bin.

Accumulating events with setting limits will reduce processing time and benefit for using min-max normalization, contributing to more stable training. As shown in **Algorithm 1**, fast causal event volume offers the advantage that once events at the same coordinate reach the defined limit, they are considered to have sufficient information, and no further accumulation is performed, reducing the time to accumulate events. Fig. 3 gives examples of different event accumulation methods: event volume, causal event volume, and fast causal event volume.

B. Network

The network is designed with a focus on lightweight architecture. We use MobileNetV3 [27] as the backbone for feature extraction, taking advantage of depthwise separable convolution (DSC) blocks to reduce complexity. Furthermore, we accelerate the Feature Pyramid Network (FPN) [39] by replacing traditional convolution blocks with DSC blocks, enhancing overall performance.

1) *MobileNetV3 Backbone*: MobileNetV3 reduces model size using DSC and enhances model expressiveness with squeeze-and-excitation (SE) blocks, achieving high inference efficiency and high. MobileNetV3 has been proven to be an excellent backbone for models working on edge devices like mobile phones and XR devices.

2) *FPN with DSC*: We replace all normal convolution blocks in FPN with DSC, reducing the parameters of FPN and improving the speed of feature fusion. The FPN of

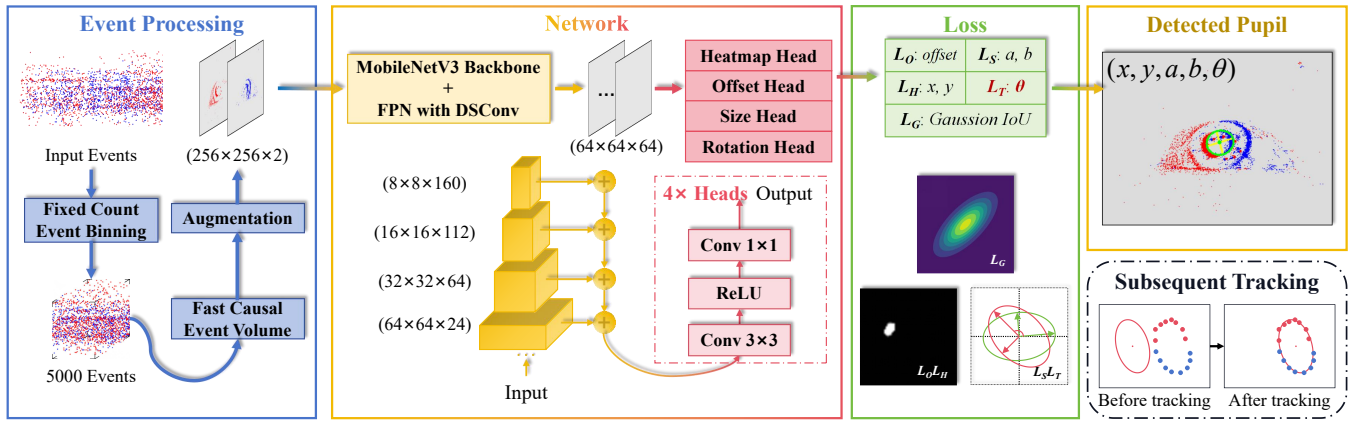


Fig. 2. Flowchart of FACET. **Event Processing:** Input events are converted to a frame-like format using fixed count binning, fast causal event volume, and augmentation for training. **Network:** A MobileNetV3 backbone with FPN and DSC extracts and fuses features, which are then passed to four heads. **Loss:** Our total loss function includes several components, among which the customized trigonometric loss L_T plays a crucial role. The term L_T specifically addresses discontinuities in angle prediction, effectively measuring the difference between the predicted ellipse and the ground truth when combined with other losses. **Detected Pupil:** FACET directly generates ellipses end-to-end, unlike segmentation networks that first obtain a mask and then fit an ellipse. **Subsequent Tracking:** This direct ellipse generation lays the foundation for high-frequency event-based eye-tracking methods [21], [37].

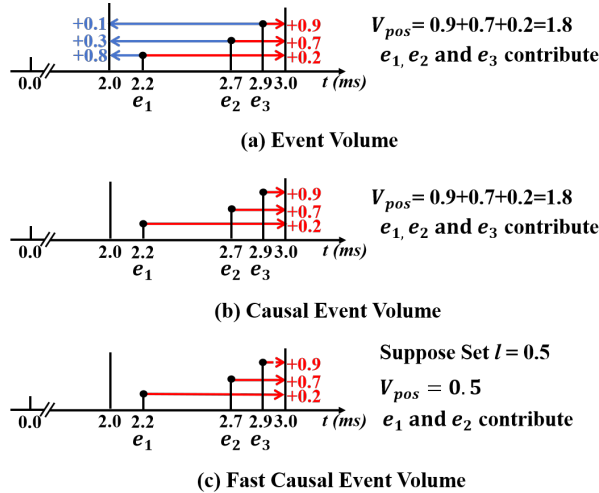


Fig. 3. Examples of different event accumulation methods: (a) Event Volume, (b) Causal Event Volume, (c) Fast Causal Event Volume. We consider accumulating the events at 3.0 ms timestamp, three events e_1, e_2, e_3 with positive polarity occur at 2.2 ms, 2.7 ms and 2.9 ms respectively. Since event volume (a) does not have temporal causality, these events will also affect the result at 2.0 ms, meaning that future events will influence past time. In the causal event volume example (b), temporal causality is preserved, and all events within the time window are processed. Our proposed fast causal event volume example (c) introduces a limit $l = 0.5$ to optimize the accumulation. This reduces the contribution of earlier events (like e_1), speeding up the process for real-time inference, where only e_1, e_2 contribute based on the defined limit.

FACET can be represented as (5):

$$P_i = \text{DSC}(C_i) + \text{Upsample}(P_{i+1}), \quad i \in \{5, 4, 3, 2\} \quad (5)$$

where C_i represents the feature maps from different stages of MobileNetV3, and P_i represents the corresponding feature maps in the FPN.

DSC blocks consist of depthwise convolution (DWC) followed by a 1×1 convolution, extracting features space-wise and channel-wise respectively. DSC reduces computa-

tional complexity from $O(HWC^2)$ to $O(HWC + C^2)$. The final feature map P_2 with $(64, 64, 64)$ dimension is used in procedures afterward.

3) *Heads:* The output feature map of the FPN is processed by four detection heads: heatmap head, offset head, size head and rotation head. Each head is composed of a 3×3 convolution, ReLU, and a 1×1 convolution, as shown in fig. 2. The heatmap head outputs a 64×64 heatmap which is used to predict the center of the pupil. The location with the maximum value in the heatmap is considered to be the center of the ellipse. Then an offset of the center is predicted by the offset head to refine the ellipse center, compensating for the quantization error from the limited resolution. The size head predicts the major and minor axis (a, b) of the ellipse. The rotation head predicts the rotation of the ellipse. Raw rotation predictions are presented in the form of $\vec{r} = (\sin(2\theta), \cos(2\theta))$. Then we normalize the prediction $\vec{r} = \hat{r} / \|\hat{r}\|_2$ and use \vec{r} to recover the rotation θ of the ellipse. The reason we choose this format of rotation representation is explained in IV-C.1.

C. Loss

In FACET, we designed a comprehensive loss function to enhance the model performance, defined as follows:

$$L = \lambda_H L_H + \lambda_O L_O + \lambda_S L_S + \lambda_G L_G + \lambda_T L_T \quad (6)$$

where L_H is the Heatmap Loss, L_O is the Offset Loss, L_S is the Size Loss, and L_G is the Gaussian IoU Loss, and L_T is the Trigonometric Loss. The innovative aspect of our approach is the introduction of Trigonometric Loss (L_T), which significantly improves angle prediction by addressing the discontinuity in traditional angle loss computation.

1) *Trigonometric Loss:* We define the rotation of an ellipse as placing the major axis of the ellipse horizontally and rotating the ellipse around its center at an angle $\theta \in [0^\circ, 180^\circ)$. Due to the symmetry of ellipses and the

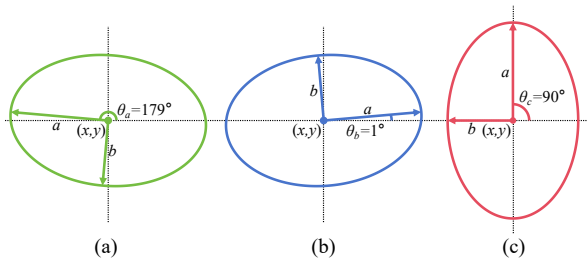


Fig. 4. Examples of ellipses at different angles. (a) $\theta_a = 179^\circ$, (b) $\theta_b = 1^\circ$, and (c) $\theta_c = 90^\circ$. Although 179° and 1° differ numerically, they produce ellipses more similar to each other than to 90° , implying their corresponding loss should reflect this pattern.

TABLE I
COMPARISON OF L_T AND L_A BETWEEN DIFFERENT ANGLES

	θ_a, θ_b	θ_a, θ_c	θ_b, θ_c
L_A	3.1067	1.5533	1.5533
L_T	0.0049	3.9988	3.9988

periodicity of rotations, 0° and 180° represent the same ellipse, which means the two ends of the range $[0, 180)$ should be continuous. Regular loss functions usually measure the norm of the difference between the prediction and the ground truth, leading to a huge discontinuity at the two ends. This discontinuity results in a large gradient event if the real difference between the prediction and the ground truth is small. The mismatch harms the training of the model.

To deal with this discontinuity, we propose Trigonometric Loss. The model predicts $\hat{r}_p = (\sin(\hat{2\theta}), \cos(\hat{2\theta}))$. The trigonometric loss L_T calculates the L2 loss between \hat{r}_p and the ground truth \vec{r}_g :

$$L_T = L_2(\hat{r}_p, \vec{r}_g) \quad (7)$$

This mapping from θ to $(\sin(2\theta), \cos(2\theta))$ transfers the discontinuous domain $[0, 180)$ to a continuous 2D domain. For example, in Fig. 4, $\theta_a = 179^\circ$ should be similar to $\theta_b = 1^\circ$. Therefore, the loss should be small. But $\theta_c = 90^\circ$ should be very different from θ_a and θ_b , thus should have a big loss. As is shown in Table I, if we compare L_T with regular L1 angle loss $L_A = L_1(\theta_p, \theta_g)$, we can see that $L_A(\theta_a, \theta_b)$ is even bigger than $L_A(\theta_a, \theta_c)$ and $L_A(\theta_b, \theta_c)$. In contrast, L_T offers a more reasonable loss, where $L_T(\theta_a, \theta_b) \simeq 0$ and $L_T(\theta_a, \theta_c) = L_T(\theta_b, \theta_c) \gg L_T(\theta_a, \theta_b)$.

2) *Other Loss Components:* Beyond the proposed Trigonometric Loss, the total loss function incorporates the Heatmap Loss L_H , Offset Loss L_O , and Size Loss L_S from CenterNet [40] and Gaussian IoU Loss L_G from EIDet [41]. L_H is the focal loss of the heatmap. L_O and L_S are smooth L1 losses between the predicted offset, scale and their corresponding ground truth. Gaussian IOU loss L_G is proposed in EIDet. An ellipse bounding box $B(x, y, a, b, \theta)$ can be reformulated in a 2D Gaussian distribution $G(\mu, \Sigma)$. L_G is measured using the wasserstein distance [42] between the predicted distribution and the ground truth distribution.

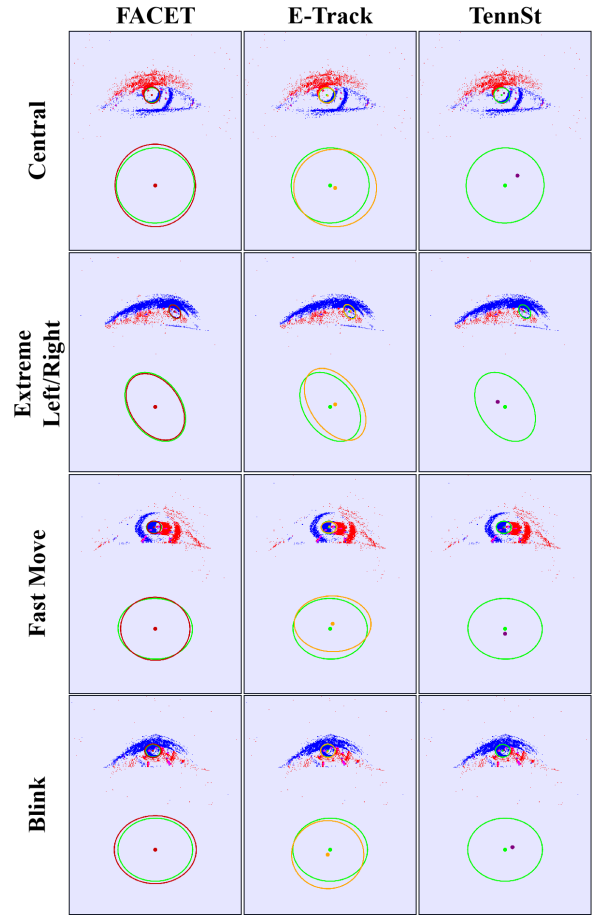


Fig. 5. Visual comparison of E-Track, TennSt, and our FACET in four typical scenarios.

V. EXPERIMENTAL RESULTS

We evaluate our model and other comparative models: E-track [21], EV-Eye [20], EIDet [41], and TennSt [19], on the enhanced EV-Eye dataset described in Section III. All metrics are obtained at a resolution of 64×64 .

A. Training Details

We train our models implemented with PyTorch [43] on a single NVIDIA RTX 3090 GPU. We used a batch size of 32, with 70 training epochs. The optimizer is Adam, with an initial learning rate of 1×10^{-3} and a weight decay of 1×10^{-5} . For the first five epochs, the warm-up learning rate is 1×10^{-5} , and the learning rate will decay by a factor of 0.7 every 10 epochs thereafter. For the proposed fast causal event volume, we set the limit l to be 25. We apply data augmentation techniques such as rotation, scaling, translation, and horizontal flip to simulate varying distances, angles, positions, and orientations of the eye relative to the event camera, improving model generalization.

B. Accuracy Results

In Table II, P_n ($n \in \{10, 5, 1\}$) represents the probability that the predicted pupil center is within n pixels of the true center, and Pixel Error (PE) represents the average

TABLE II
COMPARISON OF ACCURACY, PARAMETERS, GFLOPS, AND INFERENCE TIME. THE BEST METRIC IS IN **BOLD**, AND THE SECOND BEST IS UNDERLINED.

Method	P_{10} (%)	P_5 (%)	P_1 (%)	PE (pixel)	Params	GFLOPs	Inf. Time (ms)
E-Track	99.17	98.28	79.22	1.6680	17.27 M	40.19	0.9443
EV-Eye	99.92	99.91	98.87	0.3231	17.27 M	40.11	0.9438
EIDet	99.76	99.48	95.32	0.6273	16.82 M	8.30	*12.3854
TennSt	98.55	96.77	73.67	1.1291	0.81 M	<u>5.49</u>	0.3384
FACET (ours)	100	99.98	99.59	0.2030	<u>3.92 M</u>	3.44	<u>0.5302</u>

*The inference time for EIDet is obtained using PyTorch, as EIDet includes a custom module that is incompatible with official TensorRT.

TABLE III
ABLATION STUDY RESULTS

	P_1 (%)	PE (pixels)	EPT (ms)
FACET (Ours)	99.59	0.2030	1.6493
Event Accumulating Method - Fast causal event volume			
-Causal event volume	99.59	0.2193	1.7799
-Event volume	98.81	0.2500	1.8502
Event Binning Method - Fixed Count 5000 evts			
-500 evts	93.52	0.4326	0.8311
-1000 evts	96.43	0.3147	1.0064
-2000 evts	98.43	0.2453	1.2894
-10000 evts	99.89	0.2194	2.8983
-10000 μ s	97.29	0.2886	0.8480
Loss - Trigonometric Loss			
-Angle Loss	98.90	0.2878	-

distance from the predicted pupil center to the true center, measured in pixels. It can be seen that FACET achieved the best performance in all metrics, with a 0.2030-pixel error, far surpassing other methods. This is reflected in the visualized results in Fig. 5. We selected four typical scenarios: central, extreme right/left, fast move, and blink, and compared FACET with E-Track and TennSt. The visual results proved that the FACET achieves the highest accuracy.

C. Efficiency Results

Efficiency is evaluated using three metrics: model parameters, number of operations and inference time, with the latter measured per sample, accelerated by TensorRT on an RTX 3090. As shown in Table II, FACET outperforms other models in both the number of operations (3.44 GFLOPs) and the inference time (0.5302 ms), while also having the second smallest parameter count (3.92 M). E-Track and EV-Eye have similar parameter counts, GFLOPs, and inference times. Compared to FACET, they require 4.4 \times and 11.7 \times more parameters and arithmetic operations, respectively, and their inference time is 1.8 \times longer than that of FACET. EIDet has 4.3 \times more parameters and 2.4 \times more GFLOPs than FACET. TennSt, a fully convolutional network, has the lowest parameter count (0.81 M) and the fastest inference time (0.3384 ms). However, it requires 5.49 GFLOPs, 1.6 \times more than FACET, and is incompatible with the subsequent ellipse-based tracking module. Furthermore, TennSt has suboptimal accuracy, achieving a P_1 score of 73.67%.

D. Ablation Studies

Table III presents the results of the ablation study. EPT denotes the event processing time in milliseconds. FACET performs well across all metrics, achieving a P_1 of 99.59%, a PE of 0.2030 pixels, and an EPT of 1.6493 ms. For event accumulation, we found that traditional event volume and causal event volume methods increased the EPT by 0.20 ms and 0.13 ms, respectively. In contrast, using fast causal event volume leads to a decrease in all metrics. For the event binning method, fixing the event count to 5000 provides the best-balanced performance. Reducing the count to 500 reduces P_1 to 93.52% and increases PE to 0.4326 pixels, indicating that fewer events do not capture enough information. Increasing the count to 10,000 raises P_1 to 99.89%, but also increases the EPT to 2.8983 ms, nearly 1.8 \times longer than using 5000 events. Using a fixed time interval reduces the EPT to 0.8480 ms, but with variable event counts, the accuracy drops to P_1 of 97.29%. Using Angle Loss instead of Trigonometric Loss lowers P_1 to 98.90% and increases PE to 0.2878 pixels, confirming that our loss design improves accuracy.

VI. CONCLUSION

This work enhances the existing event-based eye-tracking dataset EV-Eye and proposes a fast and accurate eye-tracking solution: FACET, using pure event data. FACET directly outputs ellipses accurately and quickly for subsequent tracking. It uses fast causal event volume to reduce event processing time and a novel trigonometric loss to address the discontinuity in traditional angle prediction. Our experiments demonstrate that FACET is competitive in efficiency while achieving superior accuracy among the state-of-the-art methods, which highlights FACET's significant potential for eye tracking in XR environments. In future work, we aim to integrate FACET into an optimized XR system using neural processing units and event-based sensors, enabling seamless real-time eye tracking on headsets.

VII. ACKNOWLEDGMENT

Thank you to Weining Ren for the valuable suggestions provided for this article.

REFERENCES

- [1] A. Plopski, T. Hirzle, N. Norouzi, L. Qian, G. Bruder, and T. Langlotz, "The eye in extended reality: A survey on gaze interaction and eye tracking in head-worn extended reality," *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1–39, 2022.
- [2] X. Jin, S. Chai, J. Tang, X. Zhou, and K. Wang, "Eye-tracking in ar/vr: A technological review and future directions," *IEEE Open Journal on Immersive Displays*, 2024.
- [3] P. Apple, "Introducing apple vision pro: Apple's first spatial computer," <https://www.apple.com/newsroom/2023/06/introducing-apple-vision-pro/>, 2023, accessed: 2024-01-13.
- [4] A. S. Fernandes, T. S. Murdison, and M. J. Proulx, "Leveling the playing field: A comparative reevaluation of unmodified eye tracking as an input and interaction modality for vr," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 5, pp. 2269–2279, 2023.
- [5] Z. Hu, K. Zhao, B. Zhou, H. Guo, S. Wu, Y. Yang, and J. Liu, "Gaze target estimation inspired by interactive attention," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 12, pp. 8524–8536, 2022.
- [6] I. B. Adhanom, P. MacNeilage, and E. Folmer, "Eye tracking in virtual reality: a broad review of applications and challenges," *Virtual Reality*, vol. 27, no. 2, pp. 1481–1505, 2023.
- [7] M. Yang, Y. Gao, L. Tang, J. Hou, and B. Hu, "Wearable eye-tracking system for synchronized multimodal data acquisition," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.
- [8] N. Menéndez González and E. Bozkir, "Eye-tracking devices for virtual and augmented reality metaverse environments and their compatibility with the european union general data protection regulation," *Digital Society*, vol. 3, no. 2, p. 39, 2024.
- [9] K. Kurzhals, M. Hlawatsch, C. Seeger, and D. Weiskopf, "Visual analytics for mobile eye tracking," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 301–310, 2016.
- [10] P. Verghese and B. R. Beutter, "Motion processing," *Encyclopedia of the Human Brain*, pp. 117–135, 2002.
- [11] S. Doroudian and Z. Wartell, "Collaboration in immersive environments: Challenges and solutions," *arXiv preprint arXiv:2311.00689*, 2023.
- [12] N. Stein, D. C. Niehorster, T. Watson, F. Steinicke, K. Rifai, S. Wahl, and M. Lappe, "A comparison of eye tracking latencies among several commercial head-mounted displays," *i-Perception*, vol. 12, no. 1, p. 2041669520983338, 2021.
- [13] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 db 15 μs latency asynchronous temporal contrast vision sensor," *IEEE journal of solid-state circuits*, vol. 43, no. 2, pp. 566–576, 2008.
- [14] Q. Chen, Z. Wang, S.-C. Liu, and C. Gao, "3et: Efficient event-based eye tracking using a change-based convlstm network," in *2023 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2023, pp. 1–5.
- [15] P. Bonazzi, S. Bian, G. Lippolis, Y. Li, S. Sheik, and M. Magno, "Retina: Low-power eye tracking with event camera and spiking hardware," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5684–5692.
- [16] N. Li, M. Chang, and A. Raychowdhury, "E-gaze: Gaze estimation with event camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 7, pp. 4796–4811, 2024.
- [17] A. N. Angelopoulos, J. N. Martel, A. P. Kohli, J. Conradt, and G. Wetzstein, "Event based, near eye gaze tracking beyond 10,000 hz," *arXiv preprint arXiv:2004.03577*, 2020.
- [18] Z. Wang, C. Gao, Z. Wu, M. V. Conde, R. Timofte, S.-C. Liu, Q. Chen, Z.-J. Zha, W. Zhai, H. Han *et al.*, "Event-based eye tracking. ais 2024 challenge survey," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5810–5825.
- [19] Y. R. Pei, S. Brüers, S. Crouzet, D. McLelland, and O. Coenen, "A lightweight spatiotemporal network for online eye tracking with event camera," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5780–5788.
- [20] G. Zhao, Y. Yang, J. Liu, N. Chen, Y. Shen, H. Wen, and G. Lan, "Ev-eye: Rethinking high-frequency eye tracking through the lenses of event cameras," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [21] N. Li, A. Bhat, and A. Raychowdhury, "E-track: Eye tracking with event camera for extended reality (xr) applications," in *2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 2023, pp. 1–5.
- [22] Z. Wang, Z. Wan, H. Han, B. Liao, Y. Wu, W. Zhai, Y. Cao, and Z.-j. Zha, "Mambapupil: Bidirectional selective recurrent model for event-based eye tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5762–5770.
- [23] T. Stoffregen, H. Daraei, C. Robinson, and A. Fix, "Event-based kilohertz eye tracking using coded differential lighting," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2515–2523.
- [24] X. Lin, H. Ren, and B. Cheng, "Fapnet: An effective frequency adaptive point-based eye tracker," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5789–5798.
- [25] A. G. Howard, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
- [26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.
- [27] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.
- [28] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.
- [29] C.-C. Lai, S.-W. Shih, and Y.-P. Hung, "Hybrid method for 3-d gaze tracking using glint and contour features," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 1, pp. 24–37, 2014.
- [30] C. Mestre, J. Gautier, and J. Pujol, "Robust eye tracking based on multiple corneal reflections for clinical applications," *Journal of biomedical optics*, vol. 23, no. 3, pp. 035001–035001, 2018.
- [31] T. Pfeiffer and C. Memili, "Model-based real-time visualization of realistic three-dimensional heat maps for mobile eye tracking and eye tracking in virtual reality," in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, 2016, pp. 95–102.
- [32] P. L. Mazzeo, D. D'Amico, P. Spagnolo, and C. Distanto, "Deep learning based eye gaze estimation and prediction," in *2021 6th International Conference on Smart and Sustainable Technologies (SpliTech)*. IEEE, 2021, pp. 1–6.
- [33] J. Kim, M. Stengel, A. Majercik, S. De Mello, D. Dunn, S. Laine, M. McGuire, and D. Luebke, "Nvgaze: An anatomically-informed dataset for low-latency, near-eye gaze estimation," in *Proceedings of the 2019 CHI conference on human factors in computing systems*, 2019, pp. 1–12.
- [34] K. I. Lee, J. H. Jeon, and B. C. Song, "Deep learning-based pupil center detection for fast and accurate eye tracking system," in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*. Springer, 2020, pp. 36–52.
- [35] N. Zdarsky, S. Treue, and M. Esghaei, "A deep learning-based approach to video-based eye tracking for human psychophysics," *Frontiers in human neuroscience*, vol. 15, p. 685830, 2021.
- [36] O. Deane, E. Toth, and S.-H. Yeo, "Deep-saga: a deep-learning-based system for automatic gaze annotation from eye-tracking data," *Behavior Research Methods*, vol. 55, no. 3, pp. 1372–1391, 2023.
- [37] G. Zhao, Y. Yang, J. Liu, N. Chen, Y. Shen, H. Wen, and G. Lan, "EV-Eye: Rethinking High-frequency Eye Tracking through the Lenses of Event Cameras," *Advances in Neural Information Processing Systems*, vol. 36, pp. 62169–62182, Dec. 2023.
- [38] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 989–997.
- [39] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.
- [40] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Center-net: Keypoint triplets for object detection," in *Proceedings of the*

IEEE/CVF international conference on computer vision, 2019, pp. 6569–6578.

- [41] T. Wang, C. Lu, M. Shao, X. Yuan, and S. Xia, “EIDet: An Anchor-free General Ellipse Object Detector,” in *Proceedings of the Asian Conference on Computer Vision*, 2022, pp. 2580–2595.
- [42] V. M. Panaretos and Y. Zemel, “Statistical aspects of wasserstein distances,” *Annual review of statistics and its application*, vol. 6, no. 1, pp. 405–431, 2019.
- [43] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” *Advances in neural information processing systems*, vol. 32, 2019.