

Event-Based Eye Tracking. 2025 Event-based Vision Workshop

Qinyu Chen^{1✉} Chang Gao² Min Liu³ Daniele Perrone⁴ Yan Ru Pei⁵
 Zuowen Wang⁶ Zhuo Zou⁷ Shihang Tan⁷ Tao Han² Guorui Lu¹ Zhen Xu¹
 Junyuan Ding³ Ziteng Wang³ Zongwei Wu⁸ Han Han⁹ Yuliang Wu⁹
 Jinze Chen⁹ Wei Zhai⁹ Yang Cao⁹ Zheng-jun Zha⁹ Nuwan Bandara¹⁰
 Thivyaa Kandappu¹⁰ Archan Misra¹⁰ Xiaopeng Lin¹¹ Hongxiang Huang¹¹
 Hongwei Ren¹¹ Bojun Cheng¹¹ Hoang M. Truong^{12,13} Vinh-Thuan Ly^{12,13}
 Huy G. Tran^{12,13} Thuan-Phat Nguyen^{12,13}
 Tram T. Doan^{12,13}

¹ Leiden University ²Delft University of Technology ³DVSense ⁴Prophesee ⁵NVIDIA

⁶ Institute of Neuroinformatics, UZH/ETH Zurich ⁷ Fudan University ⁸ University of Würzburg

⁹ University of Science and Technology of China ¹⁰ Singapore Management University

¹¹ The Hong Kong University of Science and Technology (Guangzhou)

¹² University of Science, VNU-HCM, Ho Chi Minh City, Vietnam

¹³ Vietnam National University, Ho Chi Minh City, Vietnam

Abstract

This survey serves as a review for the 2025 Event-Based Eye Tracking Challenge organized as part of the 2025 CVPR event-based vision workshop. This challenge focuses on the task of predicting the pupil center by processing event camera recorded eye movement. We review and summarize the innovative methods from teams rank the top in the challenge to advance future event-based eye tracking research. In each method, accuracy, model size, and number of operations are reported. In this survey, we also discuss event-based eye tracking from the perspective of hardware design.

1. Introduction

With the rapid evolution of augmented and virtual reality (AR/VR) technologies, advanced by the consumer electronics industry particularly, the role of accurate and respon-

✉ Qinyu Chen (q.chen@liacs.leidenuniv.nl) is the corresponding author.

Challenge website: <https://lab-ics.github.io/3et-2025.github.io/>. Demonstration code repository: https://github.com/EETChallenge/3et_challenge_2025. Challenge Kaggle website: <https://www.kaggle.com/competitions/event-based-eye-tracking-cvpr-2025/overview>. 3ET+ Dataset: <https://www.kaggle.com/competitions/event-based-eye-tracking-cvpr-2025/data>. Event-based Vision workshop 2025 host website: <https://tub-rip.github.io/eventvision2025/>.

sive eye-tracking systems has become increasingly important. For instance, the Apple Vision Pro features an advanced eye-tracking system that uses high-speed infrared cameras and LED illuminators to monitor eye movements with remarkable precision. This technology aims for accurate gaze estimation, allowing users to interact intuitively by simply looking at objects and confirming selections with subtle hand gestures. Beyond human-computer interaction applications, eye-tracking technology is also emerging as a valuable tool in the domain of healthcare. Tasks such as gaze estimation, pupil shape tracking, and eye movement analysis offer powerful, non-invasive methods for detecting and monitoring neurological disorders, including Parkinson's and Alzheimer's diseases [15, 26, 38].

Mobile platforms are typically constrained by strict power and computing budgets, making it challenging to deploy complex software and algorithms. In addition, eye-tracking tasks demand high-frequency sensory sampling, which imposes additional burdens on hardware and data pipelines. For mobile AR/VR applications, an eye-tracking system should be lightweight to integrate seamlessly into head-mounted devices while supporting high temporal resolution and good task accuracy. This is particularly critical considering that the human eye can move at angular velocities exceeding 300°/s and accelerations up to 24,000°/s² [3], necessitating sampling rates in the kilohertz range to accurately capture the onset and dynamics of fast eye movements. However, achieving such a high frame rate is chal-

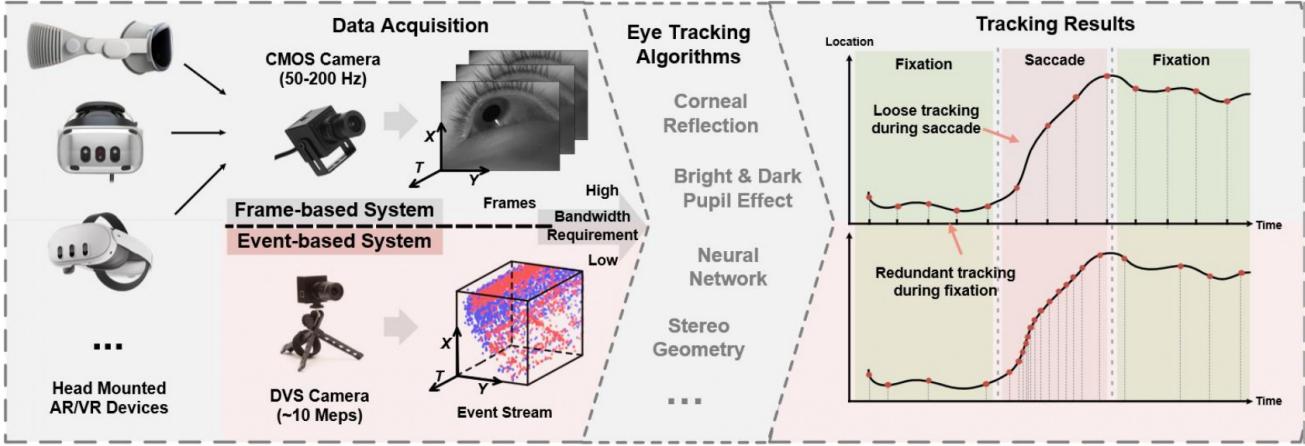


Figure 1. Comparison of the processing flow and estimation patterns between frame-based and event-based systems for eye tracking. Adapted from [45].

lenging for wearable devices, which must operate at low power levels, typically in the milliwatt range. Most head-mounted devices rely on frame-based eye-tracking systems. a recent study [43] indicates that many such systems experience tracking delays ranging from 45 to 81 ms, which is insufficient for capturing rapid eye movements that require kilohertz-level frame rates. Furthermore, frame-based sensors capable of operating at kilohertz tend to consume a large amount of power. The resulting high data throughput also demands considerable bandwidth and energy for transmission and computation, making real-time deployment on low-power wearable platforms difficult.

Event cameras, also named Dynamic Vision Sensors (DVS) [17, 28, 31], are unique vision sensors that offer several potential advantages for eye-tracking in mobile devices. Different from traditional cameras, event cameras asynchronously detect log intensity changes in brightness that exceed a certain threshold. This unique way of sensing induces spatiotemporally sparse camera outputs (events). Many research works [36, 41, 42, 46] have been proposed to exploit this spatiotemporal sparsity, aiming to reduce the hardware platform requirements of computation and energy. In addition, the temporal precision of eye-tracking tasks could benefit from the high temporal resolution data property of event cameras.

Fig. 1 shows some of the most commonly used head-mounted devices and their corresponding frame-based eye-tracking processing flow and systems in comparison to the event-based solution utilizing a DVS. The event-based approach shows promising potential in providing more robust tracking performance while requiring small bandwidth and less power consumption. These unique characteristics make event cameras highly suitable for high-speed, low-power eye tracking: they produce less data and re-

duce processing needs during fixation while still capturing fast and subtle eye movements during saccades. Previous event-based eye-tracking studies have shown promising results [3, 6, 10, 14, 29, 30, 32, 37, 44, 48, 51].

The 2025 Event-Based Eye Tracking Challenge is set to explore algorithmic potentials for event-based eye-tracking. By emphasizing efficient methods that can extract eye-position-relevant information from sparse event streams, the challenge seeks to drive advancements in event-based eye-tracking that are fast and efficient and are suitable for wearable healthcare devices and real-time AR/VR applications.

2. Event-based Eye Tracking Challenge

2.1. Introduction of the 3ET+ dataset

The 3ET+ dataset [11, 47] offers a comprehensive benchmark for event-based eye-tracking research. Captured using a DVXplorer Mini [2] event camera, it features recordings from 13 participants, each contributing between 2 to 6 sessions. During each session, subjects are required to perform five distinct eye movement tasks: *random movements*, *saccades*, *text reading*, *smooth pursuits*, and *blinks*. Ground truth annotations were provided at 100 Hz, including (1) a binary label indicating the presence or absence of a blink and (2) manually labeled pupil center spatial coordinates for precise tracking.

2.2. Task description

- **Input:** Raw events (x_i, y_i, t_i, p_i) from recording eye movements. (x_i, y_i) are spatial coordinate, t_i is the timestamp and p_i is the polarity of the event e_i .
- **Task:** Predict the spatial coordinates x_i, y_i of the pupil center at the specified timestamps, matching the frequency of the ground truth, within the input space.

- **Metric:** The evaluation metric in this year’s challenge differs from that of last year [47]. This year, the primary metric on the Kaggle leaderboard is pixel error, defined as the Euclidean distance between the spatial coordinates of the predicted label and the ground truth. In contrast, last year’s leaderboard used p-accuracy as the main evaluation metric. Under this metric, a prediction is considered correct if the pixel error is within p pixels. The leaderboard used a threshold of $p = 10$ pixels. The shift from p-accuracy to pixel error was motivated by the observation that last year’s models often achieved near-perfect p₁₀ scores (close to 100%), making it difficult to distinguish between high-performing models and limiting the potential for further improvement.

2.3. Data loading and training pipeline

The challenge provided participants with a convenient data loading and training pipeline. The data loader was designed to be compatible with the Tonic library [27], allowing users to experiment with different event-based feature representations. It also supports caching of generated features either in memory or on disk during the first training epoch, enabling faster data loading in subsequent epochs. For the training process, participants could easily integrate their own deep learning models and adjust hyperparameters as needed. A machine learning monitoring library, namely the MLFlow library [1], was also provided in the challenge pipeline code for the participants to monitor various metrics and to record the hyperparameters, as well as the checkpoints.

2.4. Challenge phases

The challenge is organized into three key phases:

1. **Preparation Phase** (Before 15. Feb. 2025): Finalize the challenge dataset, develop the code pipeline, set up the competition website, and prepare the Kaggle platform.
2. **Competition Phase** (Starting 15. Feb. 2025): The Kaggle competition officially launches. Teams can register, download the dataset, and begin working on their solutions.
3. **Submission and Evaluation Phase** (15. Mar. 2025): The submission portal closes. Private leaderboard scores are revealed, and top-performing teams are invited to submit their factsheets and source code. Selected teams are also encouraged to contribute a paper detailing their methods to the associated workshop.

2.5. Related Challenge

This challenge is part of the 2025 Event-based Vision Workshop and serves as the second edition of the Event-based Eye Tracking Challenge, following the first iteration presented in [47]. The main improvements in this year’s challenge include upgrading the label frequency from 20 Hz to

100 Hz for finer temporal resolution, and changing the evaluation metric from p-accuracy to pixel error for more precise performance differentiation. The results demonstrate a notable improvement in tracking accuracy: four participating teams achieved a pixel error below 1.7, outperforming the best results from the previous year.

3. Challenge Results

We summarize the main evaluation results from the participating teams in Tab. 1. The pixel error described in Sec. 2.2 is used as the primary evaluation metric for the Kaggle competition ranking. The model size is reported in Tab. 1 and the number of operations is reported in the sections of each team.

Team	Rank	Pixel error	Param (M)
USTCEventGroup	1	1.14	7.1
EyeTracking@SMU	2	1.42	0.8
HKUSTGZ	3	1.50	3.0
CherryChums	4	1.61	0.8

Table 1. Final results from the top performing teams. Details can be found in this survey paper.

3.1. Architectures and main ideas

The methods proposed by the participating teams range from novel pre-processing techniques and custom model architecture designs to motion-aware postprocessing method. The major novelties and contributions are summarized as follows:

Modeling Short- and Long-Term Temporal Dependencies Effectively capturing both short- and long-term temporal dynamics is important for accurate event-based eye tracking. Team USTCEventGroup modeled short-term motion with Bi-GRU and, in the following, long-term dependencies using a self-attention module enhanced with bidirectional relative positional attention. Team HKUSTGZ adopted a 3D CNN for capturing the implicit short-term temporal dynamics of the eye movements, while long-term dependencies were handled by a cascade of GRU and Mamba modules.

Data Augmentation and Generalization Strategies Team CherryChums implemented a practical augmentation pipeline, including temporal shift, spatial flip, and random event deletion, simulating real-world perturbations such as motion jitter, mirroring, and sensor dropout. In addition, pretraining on an external event dataset (synthetic 3ET dataset [10]) was used by Team USTCEventGroup to improve generalization and provide stronger initialization under limited training data conditions.

Model-Agnostic Inference-Time Post-processing Blink artifacts can interrupt event streams and cause erro-

neous gaze predictions, and temporal inconsistency may lead to unstable, non-smooth gaze trajectories. Team Eye-Tracking@SMU proposed two lightweight post-processing techniques: 1) Motion-Aware Median Filtering (M2F) to ensure temporal smoothness by adaptively smoothing gaze trajectories based on motion variance. 2) Optical Flow-based Refinement (OFE) to adjusts predictions using local event motion flow to correct spatial misalignments. These steps require no retraining or model changes and can be flexibly applied to any existing model.

3.2. Participants

There were, in total, 22 user accounts registered and participated the challenge, and 4 teams with private pixel error lower than 1.7 submitted factsheets describing their methods.

3.3. Inclusiveness and fairness

To ensure inclusiveness and fairness, several initiatives were implemented during the challenge. First, the dataset and task design were carefully optimized to reduce computational demands, making participation accessible to teams with limited hardware resources. Second, a ready-to-use training and testing pipeline was provided, allowing even those with minimal experience in event-based data to quickly get started. Finally, submitting source code alongside the factsheets was mandated to guarantee the reproducibility of all results.

4. Conclusion and Outlook

Over the last two years, this series of challenges has significantly advanced the event-based eye tracking field. The participating teams demonstrated remarkable innovation through various approaches. Although teams reported basic metrics such as parameter counts and pixel error, a deeper understanding of model efficiency and computational costs remains essential to give more useful insights for hardware designers for edge AI hardware accelerators for AR/VR wearables and more. For example, we could explore metrics that capture the actual computational workload of models, such as arithmetic operations, the memory footprint of activations (feature maps) [7], and analyze the level of sparsity if the neural network is optimized to induce spatial [8, 21–23], temporal [12, 18, 33] or spatio-temporal sparsity [9, 19, 25, 34] using tools like NeuroBench [49]. Further discussion on hardware design can be seen in the Sec. 6.

Acknowledgements

This challenge was partially (e.g., dataset collection) funded by the Swiss National Science Foundation and Innosuisse BRIDGE - Proof of Concept Project (40B1-0_213731) and

the NWO (Dutch Research Council) Talent Programme Veni AES 2023 (File number 21132). The dataset collection was partially supported by the 2023 Telluride Neuromorphic Cognition Engineering Workshop. This challenge was partially supported by DVsense.

5. Challenge Teams and Methods

The following sections present an overview of the top-performing solutions from the challenge. Each method description was authored and submitted by the respective teams as part of their contribution to this survey.

5.1. Team: USTCEventGroup

*Han Han, Yuliang Wu, Jinze Chen, Wei Zhai, Yang Cao,
Zheng-jun Zha*

University of Science and Technology of China

Contact: hanh@mail.ustc.edu.cn

Description. The USTCEventGroup proposed the Bidirectional Relative Positional Attention Transformer (BRAT) method as shown in Fig. 2. This network is composed of a spatial encoder and a temporal decoder. The former utilizes a CNN to extract geometric structural features from event representations, while the latter combines a Bi-GRU block and the BRAT block to analyze temporal motion patterns and accurately localize pupil positions. In the spatial feature extraction phase, the input binary representation is initially processed by a convolutional layer with a kernel size of 3, resulting in 32 channels. Subsequently, spatial features are extracted through three additional convolutional layers that use larger kernel sizes of 7, 5, and 5. After the first two large-kernel convolutional layer, pooling layers are applied. And following the third convolutional layer, the feature maps undergo a spatial dropout layer to mitigate overfitting.

After extracting spatial information, the features are fed into bidirectional GRU blocks to capture short-term temporal patterns. They are then further processed within the BRAT architecture to model long-range dependencies. In BRAT architecture, the standard multi-head self-attention in transformer is replaced with a bidirectional variant that explicitly encodes relative temporal distances, see Fig. 3. Given a sequence of length T and h attention heads, the output of each head is computed as:

$$\text{Attention}^i = \text{softmax} \left(\frac{\mathbf{Q}^i \mathbf{K}^{i\top}}{\sqrt{d_k}} + \mathbf{B}^i \right) \mathbf{V}^i, \quad (1)$$

where $\mathbf{B}^i \in \mathbb{R}^{T \times T}$ is a relative position bias matrix designed to modulate attention weights based on temporal distance. It is decomposed into forward and backward compo-

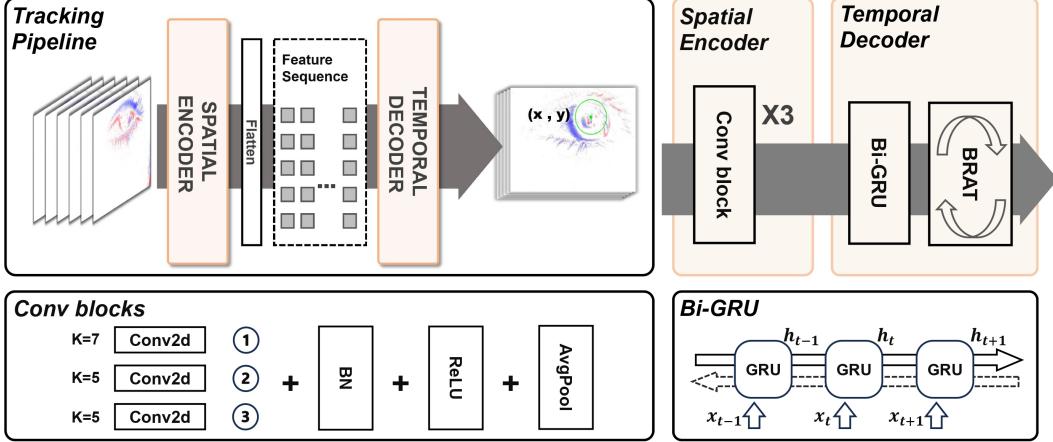


Figure 2. BRAT network by Team USTCEventGroup.

nents as:

$$\mathbf{B}_{forward}^i = \begin{cases} m^i \cdot (t - s), & t \geq s \\ 0, & t < s \end{cases}, \quad (2)$$

$$\mathbf{B}_{backward}^i = \begin{cases} 0, & t \geq s \\ m^i \cdot (s - t), & t < s \end{cases}, \quad (3)$$

where m^i denotes the sensitivity of head i to the relative position, generated via a monotonically decreasing linear mapping to progressively diminish attention to distant steps.

Furthermore, to improve the robustness of the model the USTCEventGroup adopted a multi-time-step data sampling strategy during training. Specifically, a sliding window with a fixed length was applied over the long event sequence with a stride of 1, generating dense training samples. Within each window, frames were sampled at uniform intervals defined by a step size, allowing the model to observe motion over longer temporal spans while controlling input density. For training supervision, the loss is calculated by finding the squared differences between predicted values and true labels at each time step, then taking the square root and averaging over the time dimension:

$$Loss = \frac{1}{T} \sqrt{\sum_{t=1}^T (y_{t,pred} - y_{t,label})^2}. \quad (4)$$

This formulation captures the overall error across the temporal sequence, normalizing by the sequence length to account for varying durations.

Metric	Value
Param	7.1 M
Number of MACs	2.9 G

Table 2. Model complexity of BRAT.

Implementation Details. All experiments were conducted using PyTorch, employing Cosine Annealing Warm Restart as the learning rate scheduler, starting with an initial rate of 0.001. The model was initially pretrained on the 3ET simulation dataset [10], followed by training and evaluation on the 3ET+ dataset, which took approximately 24 hours for 800 epochs at a batch size of 32 on a single RTX 2080Ti GPU. Tab. 2 shows the model complexity metrics, and the model does not require additional stages for deployment.

Results. The BRAT proposed by USTCEventGroup achieved first place on the leaderboard. The results are in the Tab. 3. The visualization results demonstrate the model's high accuracy and robust tracking capabilities in different cases. Visualization results and code are available in <https://github.com/hh-xiaohu/Event-based-Eye-Tracking-Challenge-Solution>.

p_5	p_10	p_15	pixel error
0.978	0.995	1.000	1.14

Table 3. Validation results of BRAT.

5.2. Team: EyeTracking@SMU

*Nuwan Bandara, Thivyakandappu, Archan Misra
Singapore Management University
Contact: thivyak@smu.edu.sg*

Description: Motivation & Background. In this work [5], our team specifically addresses several key limitations in the existing event-based eye tracking models. The first limitation is the handling of blink artifacts, which cause interruptions in the event data and lead to erroneous gaze

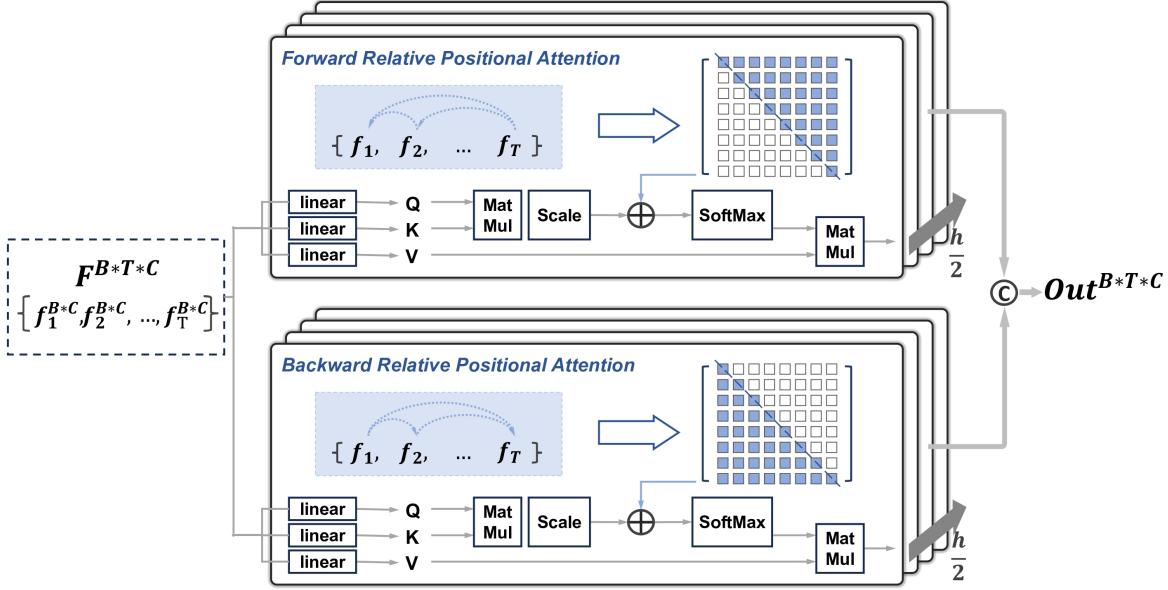


Figure 3. Bidirectional Relative Positional Attention.

predictions [50]. Another limitation is the temporal inconsistency often observed in the predictions, as eye movements are physiologically continuous and models sometimes fail to enforce this temporal smoothness, leading to abrupt gaze shifts that undermine tracking stability [4, 24]. Additionally, existing models often fail to fully leverage local event distributions, resulting in misaligned gaze predictions. These challenges, coupled with the inherent label sparsity of event datasets, make it difficult to develop a universally robust event-based tracking system.

To address these challenges, we propose a model-agnostic inference-time post-processing to enhance the accuracy and robustness of event-based eye tracking. Our approach targets the shortcomings of existing spatio-temporal models by introducing lightweight, post-processing techniques that can be integrated with any model without requiring retraining or architectural changes. This makes our method flexible and easily applicable to a wide range of existing models. The post-processing framework consists of two key components: (i) motion-aware median filtering (M2F), which enforces temporal smoothness by taking advantage of the continuous nature of eye movements, and (ii) optical flow-based local refinement (OFE), which improves spatial consistency by aligning gaze predictions with dominant motion patterns in the local event neighborhood. These refinements not only mitigate blinking artifacts but also ensure that gaze predictions remain temporally continuous and spatially accurate, even in the presence of rapid eye movements or motion artifacts.

Algorithm 1 Motion-aware median filtering

Require: Original predictions $\{x_{pred}, y_{pred}\}$, base window for local motion variance estimation w_{base} , minimum allowed smoothing window w_{min} , maximum allowed smoothing window w_{max} , percentile to determine adaptive window size p , method $f(\cdot) \in \{\text{displacement, velocity, acceleration, covariance, frequency}\}$

- 1: Output: filtered predictions $\{x_{(f,pred)}, y_{(f,pred)}\}$
- 2: local motion variance $\leftarrow f(\{x_{pred}, y_{pred}\}, w_{base})$
- 3: smoothed variance \leftarrow rolling mean(w_{base} , local motion variance)
- 4: median window \leftarrow clipping(smoothed variance, w_{min}, w_{max})
- 5: adaptive windows \leftarrow clipping(w_{min}, w_{max} , rolling(median window, w_{base}, p))
- 6: $\{x_{(f,pred)}, y_{(f,pred)}\} \leftarrow$ rolling median($\{x_{pred}, y_{pred}\}$, adaptive windows)

Description: Inference-time Post-processing. As discussed above, to address the shortcomings of existing methods in the inference stage, we propose to add two lightweight post-processing techniques specifically targeting the following limitations: (1) motion-aware median filtering (algorithm 1) to (a) ensure the temporal consistency of the predictions since the eye movements are physiologically bound to be continuous in spatial domain [24] and (b) reduce the blinking artifacts and (2) optical flow estimation in the local spatial neighbourhood (algorithm 2) to smoothly shift the original predictions if the flow vector at the original prediction is unaligned with the cumulative local neighbour-

Algorithm 2 Rule-based optical flow estimation for smooth shifts

Require: Continuous event stream with N number of events $E_{i,(t,x,y,p)}^v$ where $i \in \{1, N\}$, filtered predictions $\{x_{(f,pred)}, y_{(f,pred)}\}$, scaling parameter τ , count threshold c , difference threshold γ

- 1: Output: Refined predictions $\{x_{(R,f,pred)}, y_{(R,f,pred)}\}$
- 2: timestep $\leftarrow \frac{E^v(i=N,t_{max}) - E^v(i=1,t_{min})}{|\{x_{(f,pred)}, y_{(f,pred)}\}|}$
- 3: previous timestamp $\leftarrow E^v(i=1,t_{min}) \in E_{i,(t,x,y,p)}^v$
- 4: ROI size $R \leftarrow \tau \times 10$
- 5: **for** $j, (x_{(f,pred)}^j, y_{(f,pred)}^j) \in \{x_{(f,pred)}, y_{(f,pred)}\}$ **do**
- 6: current timestamp \leftarrow previous timestamp + $(j + 1) \times \text{timestep}$
- 7: **if** $j > c$ **then**
- 8: difference in $x \leftarrow \text{absolute}(x_{(f,pred)}^j - \text{mean}(\{x_{(f,pred)}^{j-c:j}\}))$
- 9: difference in $y \leftarrow \text{absolute}(y_{(f,pred)}^j - \text{mean}(\{y_{(f,pred)}^{j-c:j}\}))$
- 10: **if** difference in $x > \tau \times \gamma \cup$ difference in $y > \tau \times \gamma$ **then**
- 11: $R \leftarrow (1 + c) \times \tau$
- 12: **else**
- 13: $R \leftarrow (1 - c) \times \tau$
- 14: **end if**
- 15: **end if**
- 16: events in ROI $\leftarrow E^v(t \in \{\text{previous timestamp}, \text{current timestamp}\}, x \in \{x_{(f,pred)}^j - R, x_{(f,pred)}^j + R\}, y \in \{y_{(f,pred)}^j - R, y_{(f,pred)}^j + R\}) \in E_{i,(t,x,y,p)}^v$
- 17: previous timestamp \leftarrow current timestamp
- 18: $n \leftarrow \text{events in ROI} -$
- 19: **if** $n > \tau \times 10$ **then**
- 20: $dx \leftarrow 0; dy \leftarrow 0$
- 21: **for** $k \in \{1, n\}$ **do**
- 22: $dx+ = \text{events in ROI}(x = k) - \text{events in ROI}(x = k - 1)$
- 23: $dy+ = \text{events in ROI}(y = k) - \text{events in ROI}(y = k - 1)$
- 24: **if** $\text{absolute}(dx) > 0 \cup \text{absolute}(dy) > 0$ **then**
- 25: $x_{(R,f,pred)}^j \leftarrow x_{(f,pred)}^j + \frac{dx}{\|dx,dy\|}$
- 26: $y_{(R,f,pred)}^j \leftarrow y_{(f,pred)}^j + \frac{dy}{\|dx,dy\|}$
- 27: **end if**
- 28: **end for**
- 29: **end if**
- 30: **end for**

hood flow direction. This (2) is specifically inspired by our empirical observations which hint that the original predictions tend to occupy a negligence towards the event motion flow in the local neighbourhood, suggesting a lack of attention to the local event distribution in the original models.

Method	M2F	OFE	pixel error (Public)	pixel error (Private)
CG [10]	✗	✗	7.914	7.922
CG [10]	✓	✓	7.494	7.504
BB [37]	✗	✗	1.431	1.500
BB [37]	✓	✗	1.408	1.466
BB [37]	✓	✓	1.382	1.423

Table 4. Evaluation Results of Team EyeTracking@SMU’s approach.

Method	M2F	OFE	Model size size	# MACs
CG [10]	✗	✗	417K	2716.419840M
CG [10]	✓	✓	417K	2716.420096M
BB [37]	✗	✗	809K	59.537664M
BB [37]	✓	✓	809K	59.537920M

Table 5. Computational complexity details of Team EyeTracking@SMU’s approach.

More descriptively, in motion-aware filtering as shown in Algorithm 1, we first estimate the local motion variance in temporal dimension (i.e., within a set time window) using a set of alternative methods including 0th to 2nd order kinetics, covariance and frequency and subsequently, assign a median-based adaptive filter windows for each set time windows such that the kernel size for median filtering is adaptive and appropriate to the background pupil movement while also ensuring the temporal consistency. In contrast, in optical flow estimation as shown in Algorithm 2, we first estimate the appropriate size for the region of interest (ROI) around the filtered prediction using the first order derivatives of x and y and then, if the number of events within the selected ROI exceeds a set threshold, we accumulate and determine the cumulative vector trajectory of the events within ROI to softly shift the filtered prediction to further refine its spatial position.

Implementation Details: Base Models. Since our proposed method is presented as a post-processing step and works in a model-agnostic fashion, we select two recent models as base models: CNN-GRU (CG) [10], and big-Brains (BB) [37], to show the impact of the proposed pipeline towards improved pupil coordinate predictions in each case. To be specific, CG is a simple convolutional gated recurrent unit architecture which specifically designed for efficient spatio-temporal modelling to predict pupil coordinates from sparse event frames, whereas BB attempts to preserve causality and learn spatial relationships using a lightweight model consisting of spatial and temporal convolutions.

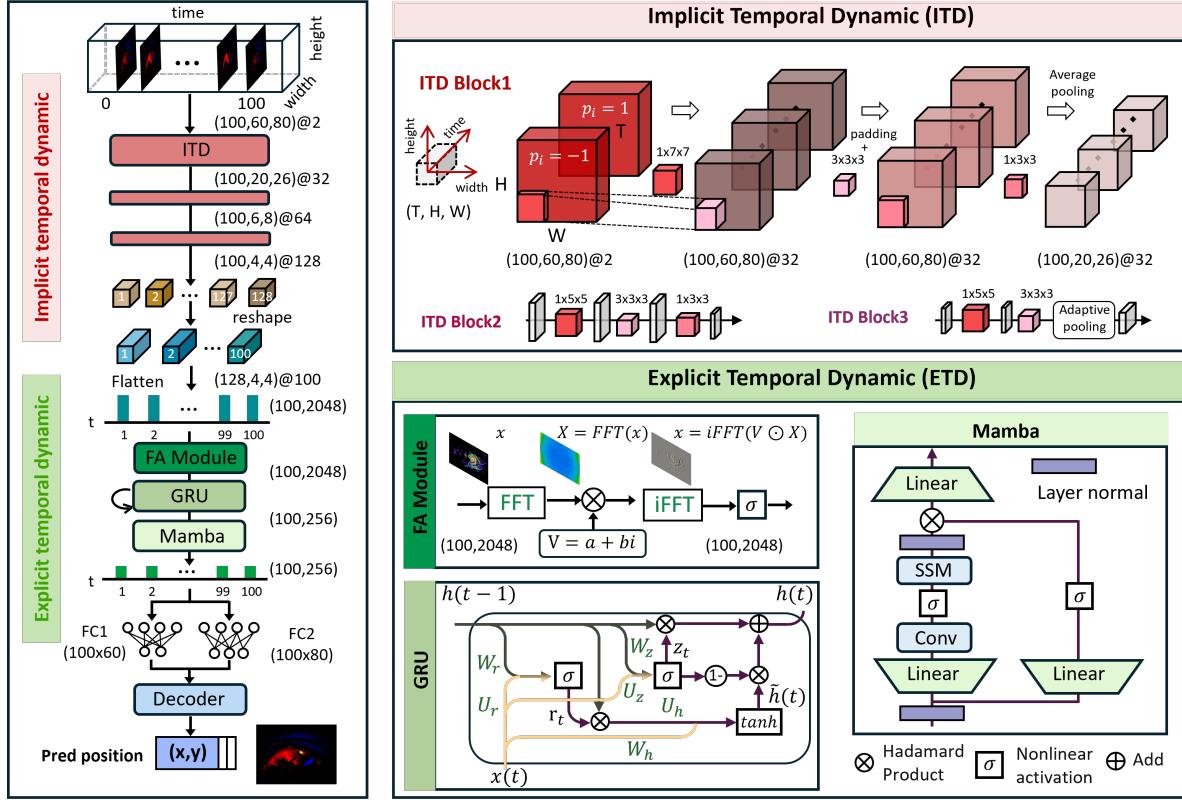


Figure 4. The architecture of TDTracker. TDTracker primarily comprises two components, Implicit Temporal Dynamic (ITD) and Explicit Temporal Dynamic (ETD), with a structure featuring three ITD components to ensure effective feature abstraction. It employs a cascaded architecture of three distinct time series models to capture temporal information comprehensively.

Results. As shown in Tab. 4, both of our post processing techniques consistently improved the results of vanilla predictions of each method and thereby suggest the efficacy of the proposed model-agnostic post-processing methods. In addition, since our methods are executed at inference time as light-weight post-processing steps, we estimate the FLOPs of M2F and OFE to be ≈ 172 and ≈ 340 per prediction instance (i.e., per event frame), respectively, whereas the learnable parameter space is effectively null. As shown in Tab. 5, we present in detail that our post-processing steps only add a negligible overhead to the base models in terms of computational complexity, despite consistently improving the vanilla prediction results of the base models. Our code is available at github/EyeLoRiN.

5.3. Team: HKUSTGZ

Xiaopeng Lin, Hongxiang Huang, Hongwei Ren, Yue Zhou,
Bojun Cheng

The Hong Kong University of Science and Technology
(Guangzhou)

Contact: bocheng@hkust-gz.edu.cn

Description. The HKUSTGZ team proposes the TD-Tracker framework [40], which is designed to address the challenges of high-speed, high-precision eye tracking using event-based cameras as shown in Fig. 4. It consists of two main components: a 3D convolutional neural network (CNN) and a cascaded structure that includes a Frequency-Aware Module [39], Gated Recurrent Units (GRU) [13], and Mamba models[20]. The 3D CNN is responsible for capturing the implicit short-term temporal dynamics of eye movements, while the cascaded structure focuses on extracting explicit long-term temporal dynamics.

In the initial phase, the event-based eye tracking data is input into the 3D CNN, which effectively captures the fine-grained short-term temporal dynamics of the eye movements. The Frequency-Aware Module is then used to enhance the model’s ability to focus on relevant frequency features that contribute to accurate eye movement tracking. Following this, the GRU and Mamba models are employed to analyze long-term temporal patterns, enabling the system to track eye movements across extended periods without losing accuracy.

Through the integration of these components, TDTracker achieves superior performance in eye movement predic-

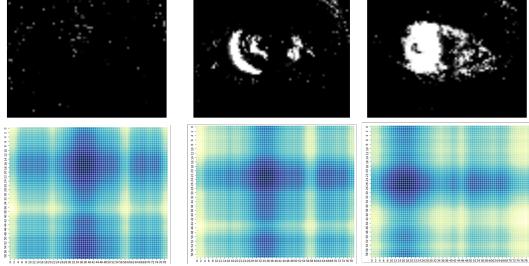


Figure 5. The visualization heatmap generated by the TDTracker.

Metric	Private	Public	Param	FLOPs	FPS
TDTracker	1.50861	1.46623	3.04M	265M	1.79 ms

Table 6. TDTracker’s performance on 3ET+ 2025.

tion and localization, enabling real-time processing with minimal computational overhead. Additionally, a prediction heatmap is generated for precise eye coordinate regression, further improving tracking accuracy. This approach demonstrates state-of-the-art performance on event-based eye-tracking challenges, showcasing the effectiveness of combining temporal dynamics with advanced neural network architectures.

Implementation Details. Our server leverages the PyTorch deep learning framework and selects the AdamW optimizer with an initial learning rate set to $2 \cdot e^{-3}$, which employs a Cosine decay strategy, accompanied by a weight decay parameter of $1 \cdot e^{-4}$. This configuration is meticulously chosen to enhance the model’s convergence and performance through adaptive learning rate adjustments. Training is conducted on an NVIDIA GeForce RTX 4090 GPU with 24GB of memory, enabling a batch size of 16.

Results. In the competition, we found that using 100 sequence training, 200 sequence testing had the highest accuracy (MSE: 1.62 to 1.55). However, since the parameters of the frequency-aware module are tied to the sequence length, we canceled this module during the competition. In addition, since our model does not consider open and closed eye cases, we simply use the ratio of the number of up and down events as the basis for judgment (set to 0.09), and when the current ratio is smaller than this value, the inference eye coordinate of the changed sample is overwritten by the inference value of the closest to this sample. What’s more, we differ from directly regressing coordinate information by using a predicted probability density map, which provides an additional probability of the model predicting this image as shown in Fig. 5. If the probability is less than 0.5, we do not believe the predicted result. Tab. 6 shows the performance of TDTracker on the private and public test sets. After post-processing, the MSE is optimized to 1.4936 on the interpolation ground truth from 3ET+ 2024.

5.4. Team: CherryChums

*Hoang M. Truong, Vinh-Thuan Ly, Huy G. Tran,
Thuan-Phat Nguyen, Tram T. Doan
University of Science, Vietnam National University Ho Chi
Minh City
Contact: 22280034@student.hcmus.edu.vn*

Description. The CherryChums team presents robust data augmentation strategies within a lightweight spatiotemporal network introduced by Pei *et al.* [37]. The network architecture is illustrated in Fig. 6 and its spatiotemporal block in Fig. 7. This approach enhances model resilience against real-world perturbations, including abrupt eye movements and environmental noise.

The data augmentation pipeline, depicted in Fig. 8, comprises temporal shift, spatial flip, and event deletion. These augmentations significantly bolster the model’s robustness while preserving computational efficiency. More specifically:

- **Temporal Shift:** Given the asynchronous nature of event-based data, temporal augmentation is crucial for improving model resilience to timing variations. We apply a random shift to event timestamps within a range of ± 200 milliseconds while ensuring proper alignment of ground truth labels. Since labels are sampled at 100Hz (every 10ms), we recompute the label indices after shifting timestamps to maintain accurate correspondence.
- **Spatial Flip:** To introduce spatial invariance, we horizontally and vertically flip the event coordinates (x, y) . The corresponding labels, including pupil center positions, undergo the same transformation to preserve spatial consistency.
- **Event Deletion:** To simulate real-world sensor noise and occlusions, we randomly remove 5% of the events while keeping the label sequence unchanged. This augmentation forces the model to learn from incomplete event streams and enhances its robustness to missing data.

The lightweight spatiotemporal network is optimized for real-time performance on edge devices, leveraging causal spatiotemporal convolutional blocks and efficient event binning. By combining these techniques, our method achieves improved accuracy and robustness in event-based eye tracking.

Implementation Details. For the lightweight spatiotemporal network, we adopt the original training configuration from Pei *et al.* [37]. Specifically, we train the model using a batch size of 32, where each batch contains 50 event frames. The network is optimized for 200 epochs using the AdamW optimizer with a base learning rate of 0.002 and a weight decay of 0.005. We employ a cosine decay learning rate schedule with linear warmup, where the

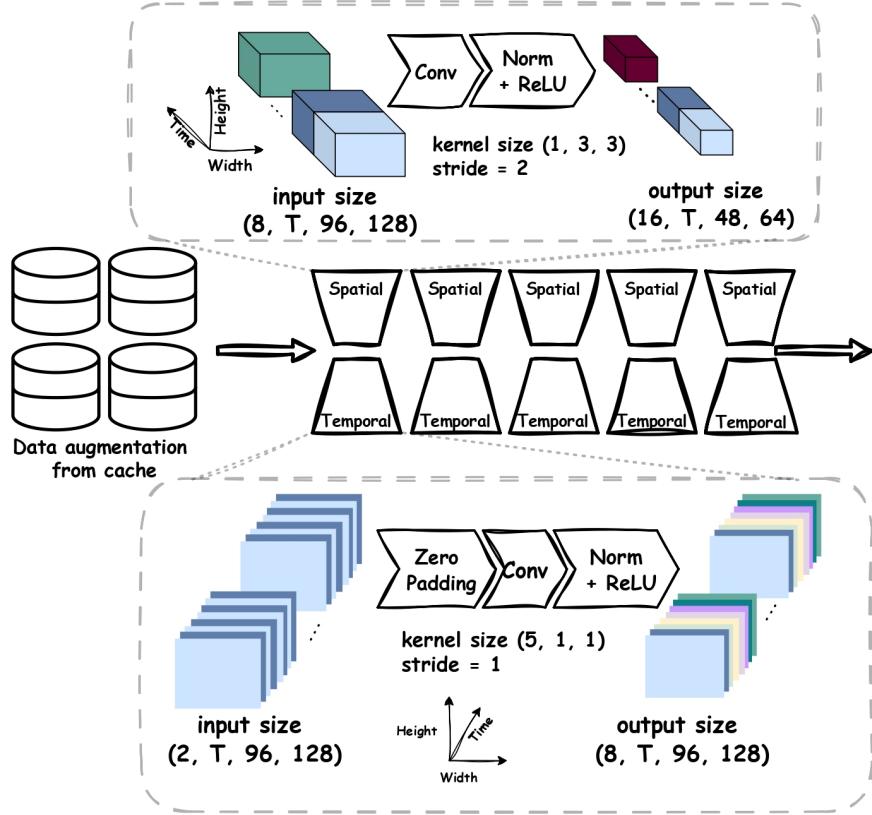


Figure 6. A compact spatiotemporal model integrating data augmentation with spatial and temporal processing blocks. Convolutional layers extract spatial and temporal features efficiently.

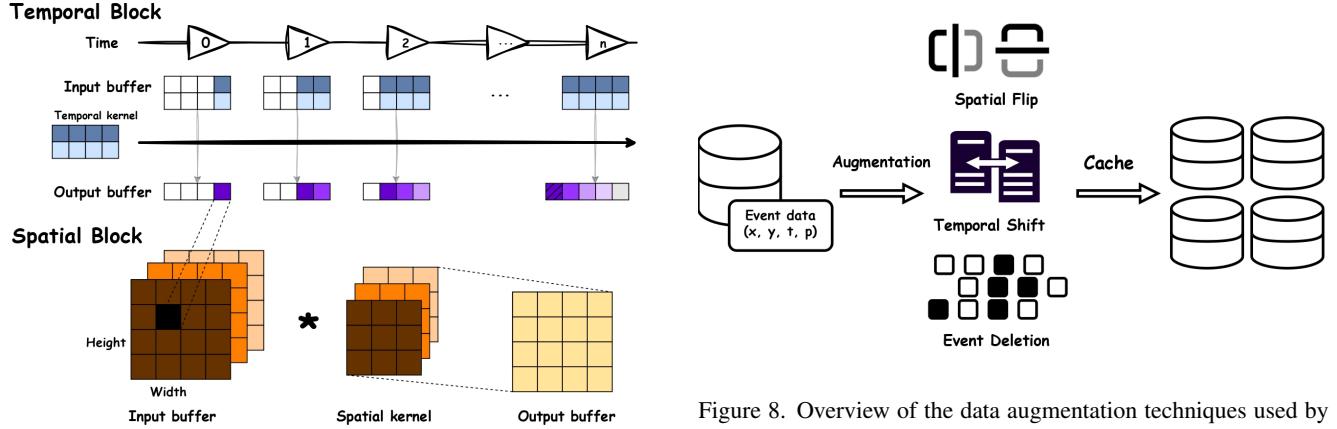


Figure 7. Illustration of spatiotemporal processing: the Temporal Block applies temporal convolution across frames, while the Spatial Block extracts spatial features using convolutional filters. warmup phase spans 2.5% of the total training steps. Additionally, we leverage automatic mixed-precision (AMP, FP16) and PyTorch compilation to accelerate training and improve efficiency. All experiments were conducted on a single NVIDIA Tesla P100 GPU provided by Kaggle.

Figure 8. Overview of the data augmentation techniques used by CherryChums.

Results. Our approach demonstrated significant improvements in robustness and accuracy, achieving a Euclidean distance error of 1.61, compared to 1.70 by the original spatiotemporal network trained without our additional augmentation strategy, as shown in Tab. 7. The results highlight the effectiveness of our data augmentation strategy in enhancing model performance under challenging conditions.

As part of our challenge report, we provide a summary

Year	Augmentation	pixel error	p_10
2024	✗	–	99.16
	✓	–	99.37
2025	✗	1.70	–
	✓	1.61	–

Table 7. Comparison of Euclidean Distance and p_10 Accuracy in both 2024 and 2025 event-based eye tracking challenges.

of the model size and the number of MACs required per frame, as shown in Tab. 8. This information is crucial for understanding the efficiency and real-time deployment capabilities of our network.

Metric	Value
Parameters	807 K
MACs per Frame	55.18 M

Table 8. Summary of Model Size and MACs

6. Hardware Discussion

The hardware design of event-based eye tracking systems necessitates careful consideration of power efficiency, latency, and adaptability to ensure robust performance in real-world applications. A central advantage of event-based architectures lies in DVS’s inherent power efficiency, which stems from its sparse data generation. Unlike conventional frame-based systems that continuously sample and transmit data, event-based systems respond only to changes in retinal activity, drastically reducing redundant data throughput. Specialized event-driven hardware designs that leverage this unique property of DVS have the potential to achieve high-performance tracking during rapid eye movements (saccades) while minimizing redundant processing during fixation periods. This efficiency can be enabled through optimization strategies such as event-triggered circuit architectures, which dynamically activate computational resources only in response to detected motion, thereby aligning power consumption with real-time demands. Additionally, this approach aligns with emerging paradigms in near-sensor or in-sensor processing, where computation is localized near the sensing element to minimize data transmission to external processors. Recent studies have showed that in-sensor event filtering and preprocessing could reduce communication bandwidth, thereby lowering both latency and power consumption [16].

A critical challenge in event-driven hardware design involves balancing the relationship between processing latency and sensor’s sampling rate. While DVS’s high sampling rate theoretically improves temporal resolution, it also brings a potential risk of overwhelming downstream pro-

cessing pipelines, leading to data congestion or potential data leakage. Therefore, optimizing buffering strategies and parallel processing architectures becomes important to handle sporadic bursts of events without introducing bottlenecks. For instance, integrating dedicated memory hierarchies or distributed processing units can prevent data contention during high-activity intervals. Such optimizations ensure that the hardware maintains low-latency responsiveness, critical for applications like foveated rendering [35] and real-time human-computer interaction, while avoiding unnecessary computational overhead during periods of eye fixation.

Moreover, configurability emerges as a pivotal design consideration to enhance the adaptability of event-based eye tracking systems across diverse operational environments. Preferably, hardware should have support for tunable parameters, such as event detection thresholds, temporal filtering windows, or region-of-interest (ROI) prioritization, to accommodate variations in lighting conditions, user ergonomics, or application-specific requirements. For example, dynamic reconfiguration of event thresholds could optimize sensitivity in low-light environments, while selective ROI processing could conserve resources by focusing computation on critical areas of the visual field. Hardware designs with this flexibility can not only extend the system’s utility across different use cases but also future-proof the design against evolving algorithmic demands. By embedding configurability into the hardware architecture, designers can strike a balance between generality and efficiency, ensuring that event-based eye tracking systems remain both practical and scalable in real-world deployments.

References

- [1] MLflow: A Machine Learning Lifecycle Platform. <https://mlflow.org/>. Accessed: 2024-04-09. 3
- [2] DVXplorer Mini User Guide. <https://inivation.com/wp-content/uploads/2023/03/DVXplorer-Mini.pdf>. 2
- [3] Anastasios N. Angelopoulos, Julien N.P. Martel, Amit P. Kohli, Jörg Conradt, and Gordon Wetzstein. Event-based near-eye gaze tracking beyond 10,000 hz. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2577–2586, 2021. 1, 2
- [4] Nuwan Bandara, Thivya Kandappu, Argha Sen, Ila Gokarn, and Archan Misra. EyeGraph: modularity-aware spatio temporal graph clustering for continuous event-based eye tracking. *Advances in Neural Information Processing Systems*, 37:120366–120380, 2024. 6
- [5] Nuwan Bandara, Thivya Kandappu, and Archan Misra. Model-agnostic inference-time post-processing and local refinement for enhanced event-based eye tracking. *arXiv*, 2025. 5
- [6] Pietro Bonazzi, Sizhen Bian, Giovanni Lippolis, Yawei Li, Sadique Sheik, and Michele 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*, pages 5684–5692, 2024. 2
- [7] Han Cai, Chuang Gan, Ligeng Zhu Massachusetts Institute of Technology, and Song Han Massachusetts Institute of Technology. Tinytl: reduce memory, not parameters for efficient on-device learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2020. Curran Associates Inc. 4
- [8] Qinyu Chen, Yan Huang, Rui Sun, Wenqing Song, Zhonghai Lu, Yuxiang Fu, and Li Li. An efficient accelerator for multiple convolutions from the sparsity perspective. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 28(6):1540–1544, 2020. 4
- [9] Qinyu Chen, Chang Gao, Xinyuan Fang, and Haitao Luan. Skydive: A spiking neural network accelerator exploiting spatio-temporal workload balance. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(12):5732–5736, 2022. 4
- [10] Qinyu Chen, Zuowen Wang, Shih-Chii Liu, and Chang Gao. 3et: Efficient event-based eye tracking using a change-based convlstm network. *2023 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2023. 2, 3, 5, 7
- [11] Qinyu Chen, Chang Gao, Min Liu, Daniele Perrone, et al. Event-Based Eye Tracking. 2025 event-based vision workshop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2025. 2
- [12] Qinyu Chen, Kwantae Kim, Chang Gao, Sheng Zhou, Taekwang Jang, Tobi Delbrück, and Shih-Chii Liu. Deltakws: A 65nm 36nj/decision bio-inspired temporal-sparsity-aware digital keyword spotting ic with 0.6v near-threshold sram. *IEEE Transactions on Circuits and Systems for Artificial Intelligence*, 2(1):79–87, 2025. 4
- [13] Junyoung Chung, Çağlar Gülcöhre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. 8
- [14] Junyuan Ding, Ziteng Wang, Chang Gao, Min Liu, and Qinyu Chen. Facet: Fast and accurate event-based eye tracking using ellipse modeling for extended reality. 2025. 2
- [15] Huiyu Duan, Guangtao Zhai, Xiongkuo Min, Zhaohui Che, Yi Fang, Xiaokang Yang, Jesús Gutiérrez, and Patrick Le Callet. A dataset of eye movements for the children with autism spectrum disorder. In *Proceedings of the 10th ACM Multimedia Systems Conference*, pages 255–260, 2019. 1
- [16] Yu Feng, Tianrui Ma, Yuhao Zhu, and Xuan Zhang. Bliss-cam: Boosting eye tracking efficiency with learned in-sensor sparse sampling. In *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*, pages 1262–1277. IEEE, 2024. 11
- [17] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022. 2
- [18] Chang Gao, Antonio Rios-Navarro, Xi Chen, Shih-Chii Liu, and Tobi Delbrück. Edgedrnn: Recurrent neural network accelerator for edge inference. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 10(4):419–432, 2020. 4
- [19] Chang Gao, Tobi Delbrück, and Shih-Chii Liu. Spartus: A 9.4 top/s fpga-based lstm accelerator exploiting spatio-temporal sparsity. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 4
- [20] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2023. 8
- [21] Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural networks. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, page 1135–1143, Cambridge, MA, USA, 2015. MIT Press. 4
- [22] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A Horowitz, and William J Dally. Eie: Efficient inference engine on compressed deep neural network. *ACM SIGARCH Computer Architecture News*, 44(3):243–254, 2016.
- [23] Song Han, Junlong Kang, Huizi Mao, Yiming Hu, Xin Li, Yubin Li, Dongliang Xie, Hong Luo, Song Yao, Yu Wang, et al. Ese: Efficient speech recognition engine with sparse lstm on fpga. In *Proceedings of the 2017 ACM/SIGDA international symposium on field-programmable gate arrays*, pages 75–84, 2017. 4
- [24] Kenneth Holmqvist, Marcus Nyström, Richard Andersson, Richard Dewhurst, Halszka Jarodzka, and Joost Van de Weijer. *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford, 2011. 6
- [25] Kevin Hunter, Lawrence Spracklen, and Subutai Ahmad. Two sparsities are better than one: unlocking the performance benefits of sparse–sparse networks. *Neuromorphic Computing and Engineering*, 2(3):034004, 2022. 4
- [26] Kwang-Hyuk Lee and Leanne M. Williams. Eye movement dysfunction as a biological marker of risk for schizophrenia. *Australian & New Zealand Journal of Psychiatry*, 34(1_suppl):A91–A100, 2000. PMID: 11129321. 1
- [27] Gregor Lenz, Kenneth Chaney, Sumit Bam Shrestha, Omar Oubari, Serge Picaud, and Guido Zarrella. Tonic: event-based datasets and transformations., 2021. Documentation available under <https://tonic.readthedocs.io>. 3
- [28] Chenghan Li, Christian Brandli, Raphael Berner, Hongjie Liu, Minhao Yang, Shih-Chii Liu, and Tobi Delbrück. Design of an rgbw color vga rolling and global shutter dynamic and active-pixel vision sensor. In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 718–721, 2015. 2
- [29] Nealson Li, Ashwin Bhat, and Arijit 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)*, pages 1–5. IEEE, 2023. 2
- [30] Nealson Li, Muya Chang, and Arijit Raychowdhury. Egaze: Gaze estimation with event camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(7):4796–4811, 2024. 2

- [31] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbrück. A 128×128 120 db $15\ \mu s$ latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008. 2
- [32] Xiaopeng Lin, Hongwei Ren, and Bojun Cheng. Fapnet: An effective frequency adaptive point-based eye tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5789–5798, 2024. 2
- [33] Shih-Chii Liu, Chang Gao, Kwantae Kim, and Tobi Delbrück. Energy-efficient activity-driven computing architectures for edge intelligence. In *2022 International Electron Devices Meeting (IEDM)*, pages 21.2.1–21.2.4, 2022. 4
- [34] Shih-Chii Liu, Sheng Zhou, Zixiao Li, Chang Gao, Kwantae Kim, and Tobi Delbrück. Bringing dynamic sparsity to the forefront for low-power audio edge computing: Brain-inspired approach for sparsifying network updates. *IEEE Solid-State Circuits Magazine*, 16(4):62–69, 2024. 4
- [35] Wenxuan Liu, Budmonde Duinkharjav, Qi Sun, and Sai Qian Zhang. Fovealnet: Advancing ai-driven gaze tracking solutions for efficient foveated rendering in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 2025. 11
- [36] Nico Messikommer, Daniel Gehrig, Antonio Loquercio, and Davide Scaramuzza. Event-based asynchronous sparse convolutional networks. In *Computer Vision – ECCV 2020*, 2020. 2
- [37] Yan Ru Pei, Sasska Brüers, Sébastien Crouzet, Douglas McLellan, and Olivier 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 Workshops*, 2024. 2, 7, 9
- [38] Elena Pretegiani and Lance M Optican. Eye movements in parkinson’s disease and inherited parkinsonian syndromes. *Frontiers in Neurology*, 8:592, 2017. 1
- [39] Hongwei Ren, Fei Ma, Xiaopeng Lin, Yuetong Fang, Hongxiang Huang, Yulong Huang, Yue Zhou, Haotian Fu, Ziyi Yang, Fei Richard Yu, et al. Frequency-aware event cloud network. *arXiv preprint arXiv:2412.20803*, 2024. 8
- [40] Hongwei Ren, Xiaopeng Lin, Hongxiang Huang, Yue Zhou, and Bojun Cheng. Exploring temporal dynamics in event-based eye tracker. *arXiv preprint arXiv:2503.23725*, 2025. 8
- [41] Alberto Sabater, Luis Montesano, and Ana C. Murillo. Event transformer. a sparse-aware solution for efficient event data processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2677–2686, 2022. 2
- [42] Y. Sekikawa, K. Hara, and H. Saito. Eventnet: Asynchronous recursive event processing. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3882–3891, Los Alamitos, CA, USA, 2019. IEEE Computer Society. 2
- [43] Niklas Stein, Diederick C Niehorster, Tamara Watson, Frank Steinicke, Katharina Rifai, Siegfried Wahl, and Markus Lappe. A comparison of eye tracking latencies among several commercial head-mounted displays. *i-Perception*, 12(1):204166952098338, 2021. 2
- [44] Timo Stoffregen, Hossein Daraei, Clare Robinson, and Alexander Fix. Event-based kilohertz eye tracking using coded differential lighting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2515–2523, 2022. 2
- [45] Shihang Tan, Jinqiao Yang, Jiayu Huang, Ziyi Yang, Qinyu Chen, Lirong Zheng, and Zhuo Zou. Toward efficient eye tracking in ar/vr devices: A near-eye dvs-based processor for real-time gaze estimation. *IEEE Transactions on Circuits and Systems I: Regular Papers*, pages 1–13, 2025. 2
- [46] Zuowen Wang, Yuhuang Hu, and Shih-Chii Liu. Exploiting spatial sparsity for event cameras with visual transformers. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 411–415, 2022. 2
- [47] Zuowen Wang, Chang Gao, Zongwei Wu, Marcos V. Conde, Radu Timofte, Shih-Chii Liu, Qinyu Chen, et al. Event-Based Eye Tracking. AIS 2024 Challenge Survey. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2024. 2, 3
- [48] Zhong Wang, Zengyu Wan, Han Han, Bohao Liao, Yuliang Wu, Wei Zhai, Yang Cao, and Zheng-jun Zha. Mamabipupil: Bidirectional selective recurrent model for event-based eye tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5762–5770, 2024. 2
- [49] Jason Yik, Korneel Van den Berghe, Douwe den Blanken, Younes Bouhadjar, Maxime Fabre, Paul Hueber, Weijie Ke, Mina A Khoei, Denis Kleyko, Noah Pacik-Nelson, et al. The neurobench framework for benchmarking neuromorphic computing algorithms and systems. *Nature Communications*, 16(1):1545, 2025. 4
- [50] Tongyu Zhang, Yiran Shen, Guangrong Zhao, Lin Wang, Xiaoming Chen, Lu Bai, and Yuanfeng Zhou. Swift-Eye: towards anti-blink pupil tracking for precise and robust high-frequency near-eye movement analysis with event cameras. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 6
- [51] Guangrong Zhao, Yurun Yang, Jingwei Liu, Ning Chen, Yiran Shen, Hongkai Wen, and Guohao Lan. Ev-eye: Rethinking high-frequency eye tracking through the lenses of event cameras. *Advances in Neural Information Processing Systems*, 36, 2024. 2