MambaPupil: Bidirectional Selective Recurrent model for Event-based Eye tracking

Zhong Wang, Zengyu Wan, Han Han, Bohao Liao, Yuliang Wu,
Wei Zhai*, Yang Cao, Zheng-jun Zha
University of Science and Technology of China
{wangzhong@mail., wanzengy@mail., hanh@mail., liaobh@mail., tronliang@mail.,
wzhai056@, forrest@, zhazj@}ustc.edu.cn

Abstract

*Event-based eye tracking has shown great promise with the high temporal resolution and low redundancy provided by the event camera. However, the diversity and abruptness of eye movement patterns, including blinking, fixating, saccades, and smooth pursuit, pose significant challenges for eye localization. To achieve a stable event-based eye-tracking system, this paper proposes a bidirectional long-term sequence modeling and time-varying state selection mechanism to fully utilize contextual temporal information in response to the variability of eye movements. Specifically, the **MambaPupil** network is proposed, which consists of the multi-layer convolutional encoder to extract features from the event representations, a bidirectional Gated Recurrent Unit (GRU), and a Linear Time-Varying State Space Module (LTV-SSM), to selectively capture contextual correlation from the forward and backward temporal relationship. Furthermore, the Bina-rep is utilized as a compact event representation, and the tailor-made data augmentation, called as **Event-Cutout**, is proposed to enhance the model's robustness by applying spatial random masking to the event image. The evaluation on the ThreeET-plus benchmark shows the superior performance of the MambaPupil, which secured the 1st place in CVPR'2024 AIS Event-based Eye Tracking challenge.*

1. Introduction

Event-based eye tracking aims to precisely predict the pupil's spatial location at any moment based on the event camera signal collected within the human eye area. It has demonstrated great importance in the field of human-computer interaction (HCI)[20], and the eye-tracking component promises the potential to be a significant functionality of future virtual reality/augmented reality (VR/AR) devices[14]. The traditional eye tracking systems rely on high-speed cameras [16], which are power-intensive and demanding for light-weight design. In contrast, the event camera provides the possibility to achieve a practical eye-tracking system with its low power consumption and high temporal resolution.

As the fastest moving organ in the human body, the state of the pupil constantly changes drastically and abruptly, which poses several hard challenges for eye tracking: 1) Loss of target due to blinking: When a person blinks, the eyelid completely covers the eyeball, leading to a brief loss of the pupil's state. Moreover, blinking generates numerous irrelevant events, which may cause non-negligible fluctuations in the model's predictions in a short period; 2) Sparse events during eye resting: As event cameras generate signals when brightness changes occur, the event information produced when the eye is at rest is extremely sparse. When the event density is insufficient to support accurate predictions, the results typically deviate and exhibit noticeable jitters; 3) Interference from other objects: These objects usually refer to glasses, eyelashes, and reflections on irises. The former generates signals when the head moves, while the latter two produce signals during eye movements. These signals are often detrimental to pupil tracking, especially the presence of glasses, which can cause significant and prolonged misalignment of the predictions.

To resolve the above challenges, digging out the contextual temporal information is obviously the key factor. Recent works try to extract the temporal correlation of pupil movement by the Recurrent Neuron Network (RNN), such as LSTM[4], to realize stable tracking. Additionally, there exist some attempts to employ a Spiking Neuron Network (SNN)[2] to track, which is tailored for event signals' asynchronous and focuses on more fine temporal relationships. However, in these works, the temporal direction is considered unidirectional, and each timestamp is taken into account equally, rendering too simple a temporal model and

*Corresponding author.

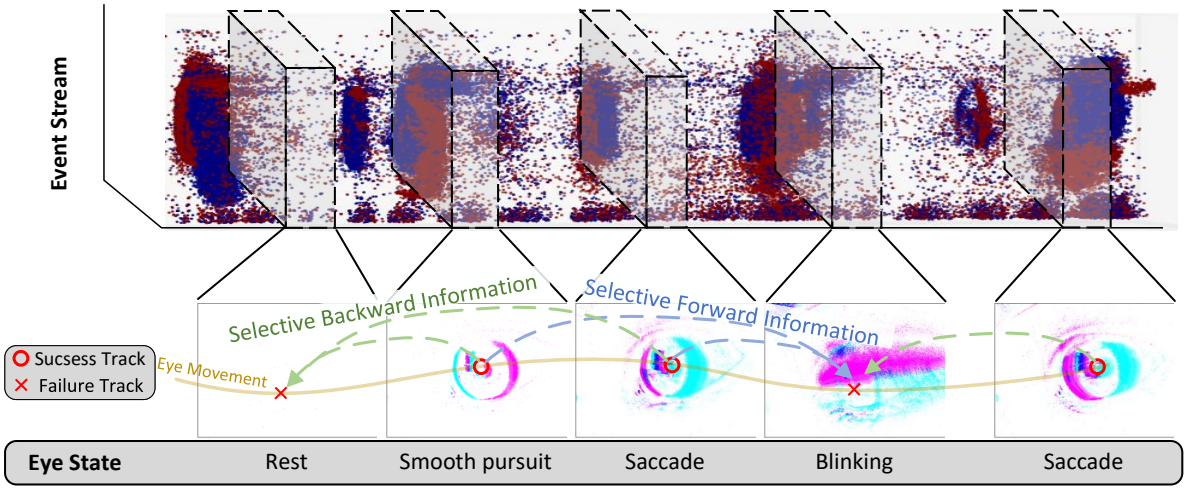

Figure 1. The diversity of eye movements, including blinking, saccades, resting, etc., poses significant challenges for accurate and stable eye tracking. To address these challenges, leveraging contextual temporal information becomes crucial in assisting with eye localization. The historical and future data of eye movements can help to infer the current position and potential trajectories of the eye.

prone to failure in challenging eye tracking.

To fully extract and utilize the temporal correlation of pupil movement, this paper attempts to model the temporal relationship bidirectionally and selectively and proposes the **MambaPupil** model for stable eye tracking. Specifically, the model utilizes the stacks of the CNN layer to extract event spatial information, which is then fed into a Dual Recurrent Module to extract the specific pupil position by contextual temporal modeling, where the bidirectional GRU is firstly employed to extract contextual information, which is input into the Linear Time-Varying State Space Module (LTV-SSM) to establish the hidden state of eye movement patterns[8].

To further improve the model's generalization performance, the Bina-rep, an event spatial-temporal binarization representation, is utilized as network input[1]. This method transforms multiple event frames accumulated within a time window into binary event graphs of specified digits, significantly reducing the input scale and avoiding interference caused by noisy events. A tailor-made data augmentation technique, **Event-Cutout**, is proposed to enhance the robustness of spatial localization by spatial random masking[12]. Experiments on challenging eye-tracking dataset demonstrate that the proposed method achieves accurate and stable tracking results even on hard samples. Moreover, due to the concise design of the model, it has low training costs while maintaining extremely fast inference speeds.

In summary, our main contributions are as follows:

- A novel framework, MambaPupil, is proposed to model the temporal relationship bidirectionally and selectively for accurate and stable eye tracking.
- The Dual Recurrent Module in MambaPupil is designed to utilize the bidirectional GRU module to rich the temporal contextual information by bidirectional temporal modeling and the LTV-SSM module to selectively cast importance into the valid eye motion stage by input-dependent weight adjusting.
- The tailored data augmentation technique, Event-Cutout, is proposed by random spatial masking, endowing the network strong robustness under challenging senses.
- Evaluation on challenging eye tracking dataset demonstrates the superior performance of the proposed approach.

2. Related Work

2.1. Eye Tracking

In eye-tracking tasks, model-based methods developed according to the structure of the eye have achieved very high accuracy[13]. These methods extract prominent geometric features of the eye from images or utilize optimization techniques to perform 3D physical modeling of the eye based on corneal and retinal reflection patterns[10, 22, 28]. However, these methods require additional equipment and frequent calibration, and they are sensitive to ambient lighting conditions. Moreover, they are susceptible to factors such as pupil color, eyelid occlusion, and eyelash occlusions[17, 25]. Appearance-based End-to-end learning methods can directly infer the position of the pupil from features of in-

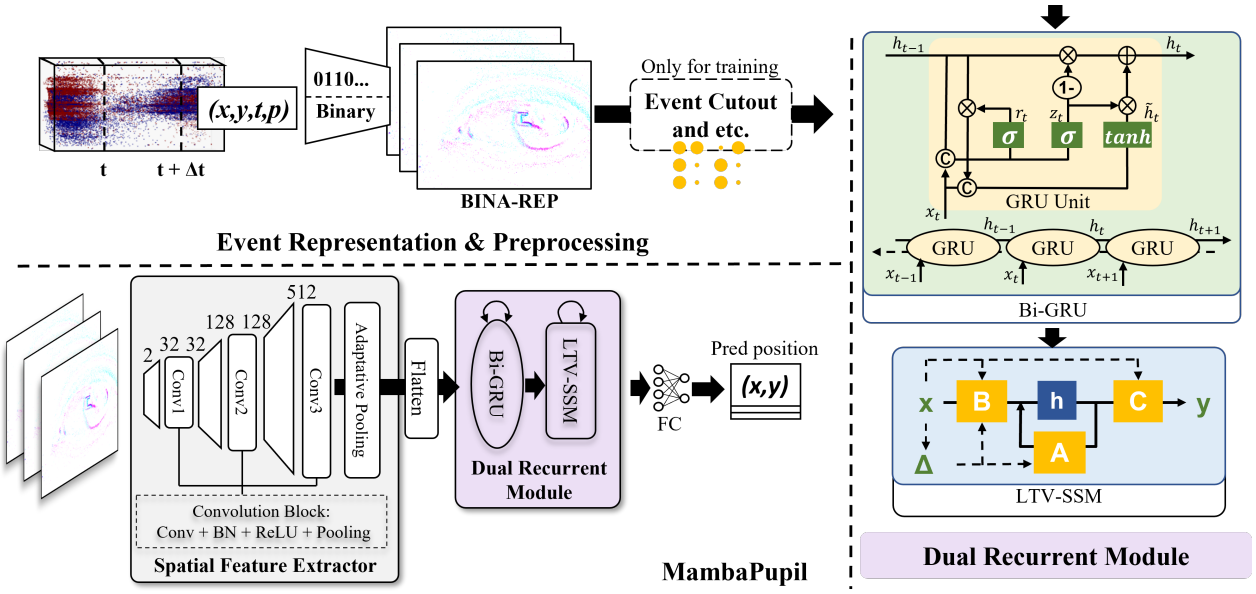

Figure 2. **Top**: Event streams are transformed into Bina-rep representations and enhanced only during the training phase. **Bottom**: MambaPupil framework is composed of the Spatial Feature Extractor, Dual Recurrent Module, and the Fully Connected layer as the final classifier. The Spatial Feature Extractor consists of stacked convolutional blocks. **Right**: The Dual Recurrent Module consists of two temporal modeling units, Bi-GRU and LTV-SSM. The Bi-GRU extracts the complete contextual information through the bi-directional temporal information flow process, and LTV-SSM achieves selective temporal state encoding through content-dependent parameter tuning.

put images extracted by CNN[3, 11, 18, 26, 27, 29]. However, the tracking speed of these methods is limited by the camera's frame rate, making it challenging to achieve sub-millisecond time resolution on wearable devices.

Event-based methods, due to the asynchronous nature of the signals, have the potential to address the time resolution issues of end-to-end models[7]. Early event-based eye tracking approaches used modeling methods as a backbone, fitting image, and event frames to parameters of the 3D physical model[21]. While this method achieves impressive model accuracy, it is still dependent on image information and significantly impacted by sensor noise. Recent purely event-based eye tracking algorithms can be divided into two directions: those based on Spiking Neural Networks (SNN) and those based on event frames. SNN-based networks[2], composed of neural cell assemblies, can fully utilize the asynchronous nature of events and are lightweight. The latter converts events into event frames with a fixed rate and extracts features related to pupil position using processing techniques similar to grayscale images[4]. Various RNN structures, such as LSTM, are commonly used to capture the temporal information of events. Aligning with existing event frame-based methods, we further improve performance through novel network design and data representing techniques.

2.2. State Space Model (SSM)

The state space model (SSM) is a concept based on control theory. It introduces a state vector and its update matrices between input and output variables, aiming to model and analyze systems more finely, allowing for more precise control of the target object. Gu et al. introduced it into the field of deep learning in [9]. They first proposed the S4 model, which captures the intrinsic features of specific tasks through the analysis of Linear Time-Invariant (LTI) systems. Subsequently, they further improved the structural simplicity and performance by introducing the S4D and S5 models. These models are utilized in a way similar to other RNN varieties to capture dependencies over different time ranges. Due to the limitations of time-invariant systems in handling all cases, in [8], they proposed the time-variant S6 model, which not only allows more parameters to be correlated with inputs but also introduces selection strategies and performance optimizations at the hardware level. In our network architecture, after generating hidden states with temporal context information using GRU, we fit a time-variant SSM to model these states, achieving excellent results in eye-tracking tasks. Recently, there have also been some works applying SSM to event-based computer vision[30].

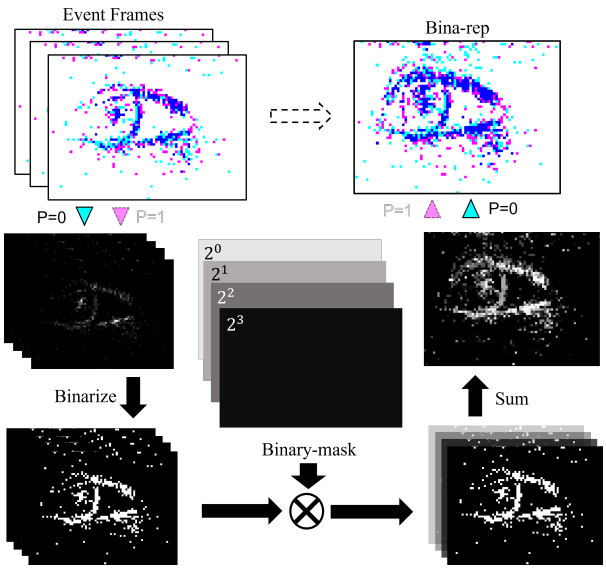

Figure 3. The Bina-rep is generated by aggregating binary-masked event frames. The routines for each polarity of event are the same and will be concatenated at last.

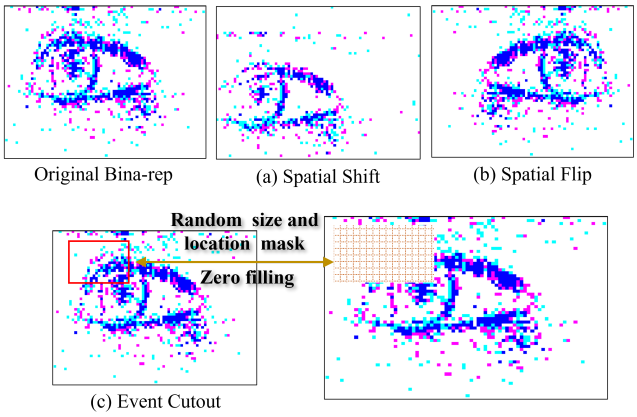

Figure 4. Various data enhancement methods are adapted to improve the robustness, including spatial flip, spatial shift, event-cutout and etc. Event-Cutout is a spatial enhancement technique suitable for event-based eye tracking, which sets the pixels within random rectangular box to zero to simulate potential external interference, *e.g.*, eye blinking.

3. Methodology

In this section, we describe the preprocessing methods of event data and augmentation techniques in Sect. 3.1. Then, we explain the structure of the proposed network in Sect. 3.2. Finally, we introduce the loss function in Sect. 3.3. The whole framework is visualized in Fig. 2.

3.1. Event Processing

Event cameras asynchronously generate event data at each pixel independently. When the logarithmic change in brightness at a pixel position (x, y) surpasses the threshold, an event is triggered as :

$$log\mathcal{I}(x, y, t) - log\mathcal{I}(x, y, t - \delta t) = pC, \qquad (1)$$

where $\mathcal{I}$ represents the scene brightness, C is the set threshold, and δt is the timestamp of the event, measured in microseconds. $p \in \{-1, 1\}$ represents the polarity of the event change, indicating either decrease or increase in brightness.

Such event points are motion-triggered in the spatiotemporal domain, forming the event stream and we transform a fixed period of event into the specific event representation for eye tracking. Next, we will detail our event representation, Bina-rep and the corresponding data augment techniques introduced in training phase.

Bina-rep Representation of Event data. Due to the similarity of spatial shape of the eye, we adopt a binarization quantization method proposed by Barchid et al.[1], which converts the event over a period of time Δt into a representation format of N-bit map, as shown in Fig. 3. Specifically, it first generates a spatially binarized event clip based on the event frames aggregated and accumulated over a period of time, after which it encodes each frame in the clip using a temporal binarization mask. And then, we accumulate the spatio-temporally binarized encoded clip to form the final Bina-rep B_e. This approach brings two benefits: 1) it reduces storage and computational costs, improving the training and inference speed of the network. The space occupied by the Bina-rep representation for the same time-length of event data is one divided by the n_{bins} in the standard event frame representation; 2) it reduces the impact of noise and redundant information on prediction results to some extent. In this representation method, isolated events are less pronounced.

Event-Cutout and Other Data Augment Techniques. To enhance the algorithm's generalization ability when dealing with challenge scenes, *e.g.* eye blinking, we utilize several event data enhancement techniques in training phase, including spatial flip, spatial shift, event-cutout and etc. These data enhancement techniques are visualized Fig. 4. And the event-cutout is a tailor-made data enhancement technique for eye tracking, of which the core idea is randomly masking spatial regions, forcing the network to learn global features and avoid getting trapped in local minutiae[12]. The event-cutout is designed by randomly sampling the height and width value as the size of spatial mask and also randomly sampling the x, y coordinate as the location of the mask location in original Bina-rep. The region in this mask will be filled into 0, indicating the event in this region will be free. This technique helps to generate the spatial interference in training phase and is particularly effective in scenarios such as blinking or wearing glasses, which hinders the model to focus on valid eye region.

3.2. Architecture of Proposed Network

In this section, we will introduce the network architecture demonstrated in Fig. 2. Overall, our network consists of two main parts: the Spatial Feature Extractor employs a convolutional encoder to extract spatial semantic features from the Bina-rep, and the Dual Recurrent Module utilizes a Bi-GRU and LTV-SSM to extract temporal constraints bidirectionally and selectively for predicting the pupil's location.

Spatial Feature Extractor. The input Bina-rep is firstly transported into the Spatial Feature Extractor, which consists of a series of convolution blocks to extract spatial pattern x_t at time t. Each convolution block is designed as following,

$$x_t = Pool(ReLU(BN(Conv(B_e)))), \tag{2}$$

where the $Conv$ is the 2d convolution operator, BN is the batch normalization layer, $ReLU$ is the relu activation function and the $Pool$ is the spatial pooling layer. To obtain spatial correlation of a broader range, larger convolutional kernels (7 or 5) are employed. After passing through three convolutional blocks, the features are fed into the global spatial pooling layer for aggregation. Subsequently, A spatial dropout[23] is adopted in the training phase, and the results are fed into the Dual Recurrent Module.

Dual Recurrent Module. In the Dual Recurrent Module, two consecutive recurrent modules are sequentially employed to extract relevant temporal information bidirectionally and selectively. The first module consists of a bidirectional Gated Recurrent Unit (GRU)[6]. It computes update and reset gate output r_t, z_t, based on inputs x_t and the previous time step's hidden state h_{t-1}, controlling the memory and forgetfulness of the hidden state. Compared with Long Short-term Memory (LSTM) network, the GRU is easier to train and less prone to issues like gradient vanishing and explosion. However, using a unidirectional network for the temporal segment may lead to non-ideal predictions at one end because of information insufficiency. To tackle this, bidirectional GRU units are employed to enhance prediction accuracy at both ends of segments, in which the sequential input will be forward and backward passed through the GRU, and the bi-directional hidden states will be concatenated as the final output. The Bi-GRU is employed as follows:

$$z_t = \sigma(W_z \cdot [h_{t\pm1}, x_t]), \tag{3}$$
$$r_t = \sigma(W_r \cdot [h_{t\pm1}, x_t]), \tag{4}$$
$$\tilde{h}_t = tanh(W \cdot [r_t * h_{t-1}, x_t]), \tag{5}$$
$$h_{t\{f,b\}} = (1 - z_t) * h_{t\pm1} + z_t * \tilde{h}_t, \tag{6}$$

The W_z, W_r, W stand for the update, rest and output gate weight respectively, σ is the sigmoid activation function, and $h_{t\{f,b\}}$ is the forward or backward hidden state. We will concatenate the forward and backward state as the bidirectional temporal information h_t.

After extracting features in the previous stage through Bi-GRU, we feed the hidden states of each time step into the LTV-SSM. We leverage the State Space Model (SSM)[8] to model the behavior patterns of eye movements selectively to cast more attention into the valid phase. Here, we first introduce the state space model, which is formalized as follows:

$$\Delta x = Ax + Bu, \tag{7}$$
$$y = Cx + Du, \tag{8}$$

where x, y, and u represent the hidden state, the output and control variables of the system, respectively, A is the state (or system) matrix, B the input matrix, C the output matrix, and D the feedforward matrix. When implementing this method as linear time varying system in deep networks, we follow the [8] strategy to treat D as fixed network parameters, and matrices A, B and C are influenced by the input sequence. Specifically, during the discretization process, we directly construct Δ, B, and C from the input sequence x, while learning the parameters of the constructor during the training process.

$$\Delta, B, C = Linear(x), \tag{9}$$
$$\Delta A = exp(\Delta * A), \tag{10}$$
$$\Delta B = \Delta * B, \tag{11}$$

Before feeding the hidden states into the SSM, root mean square normalization (RMSNorm) is applied to enhance computational efficiency. Additionally, outside the entire SSM module, we apply residual connections to mitigate the adverse effects of network complexity. Therefore, the computational process of the LTV-SSM module is as follows:

$$x_t^{'} = RMSNorm(x_t), \tag{12}$$
$$h_t = \Delta A * h_{t-1} + \Delta B * x_t^{'}, \tag{13}$$
$$y_t = C * h_t + D * x_t^{'} + x_t, \tag{14}$$

3.3. Loss Function for Training

When computing the loss function, we use the mean square error between the predicted results and the labels, averaged over each segment to avoid bringing extra complexity. Below is our loss calculation formula,

$$Loss = \sqrt{\frac{1}{L}\sum_{i=1}^{L}(y_{i,pred} - y_{i,label})^2}, \tag{15}$$

where $y_{i,pred}$ and $y_{i,label}$ represent the predicted results and training labels, respectively, for the i-th sample in the segment, and L denotes the length of each segment involved in training.

Method	train stride	p_5	p_{10}	p_{15}	p_{error}	validation loss
CNN-GRU[5, 24]	30	0.610	0.856	0.931	5.90	0.0763
	10	0.605	0.832	0.928	5.90	0.0761
	5	0.615	0.844	0.931	5.90	0.0737
CB-ConvLSTM[4]	30	0.788	0.936	0.971	3.92	0.0542
	10	0.858	0.968	0.989	5.80	0.0427
	5	0.792	0.848	0.981	3.82	0.0543
MambaPupil(Ours)	30	0.903	0.971	0.985	2.77	0.0396
	10	0.935	0.976	0.987	2.35	0.0322
	5	**0.937**	**0.984**	**0.990**	**2.03**	**0.0287**

Table 1. Comparison with other methods: p_n denotes prediction accuracy within n pixels, where higher values indicate better performance, and p_{error} represents the average prediction Euclidean distance, where lower values are preferred. We compare models trained with different training strides. It can be observed that our method consistently outperforms other methods across various training strides.

4. Experiments

4.1. Datasets and Preprocessing

We perform experiments on the EET+ dataset for training and evaluation[5, 24]. This dataset comprises 13 recorded scenarios, each containing 2-6 segments, all with an event resolution of 640x480. The motions of the human eye in these segments can be roughly categorized into five classes: random movement, saccades, reading text, blinking, and smooth pursuit. The labels for pupil positions were performed at 100Hz, and we utilized 3/4 of the data as the training set, 1/4 as the validation set, and predicted labels at a sampling rate of 20Hz following the [5] setting. The evaluation was assessed by the Euclidean distance between the predicted pupil center and the ground truth and the successful rate in which distances shorter than p pixels are regarded as successful. In this context, p_5, p_{10}, and p_{15} represent p = 5, 10, and 15, are adopt to evaluate the performance.

4.2. Implementation Details

The proposed method was implemented using PyTorch[19], and the experiments were conducted on a single NVIDIA RTX2080Ti or GTX1080Ti GPU. The training epoch was set to 1000 with a batch size of 32. All networks utilized the Adam optimizer and the Cosine Annealing Warm Restart scheduler[15], starting with a learning rate of 0.002. To improve training efficiency, the resolution of the event and label was downsampled to 80x60. The Spatial Feature Extractor in MambaPupil is composed of three convolution blocks with output channels of 32, 128, and 512, respectively, and the kernel sizes in blocks were 7, 5, and 5. As for the Bina-rep, the bit number was set to 4 empirically. Each training sequence had a length of 45 and was trained with a step size of 5; different training strides had a significant impact on the results, with smaller step sizes yielding superior performance due to generating more hard sequences, albeit at the cost of increasing training time by multiples. The following ablation and comparison experiments will mainly conducted at the train stride of 10 for efficient consideration. The data augmentation methods were conducted on 50% of the training data. As for labels for training, The dataset provides pupil position labels at 640x480@100Hz. To align with our data processing pipeline, we downsampled the time resolution to 20Hz, and spatial position of both predictions and labels are normalized.

Model	Params	Flops	Train time
CNN-GRU	37.23M	1.9T	1h42m
CB-ConvLSTM	**417.17K**	**1.09T**	2h12m
MambaPupil	8.59M	2.61T	**1h31m**

Table 2. Evaluated on a single RTX2080Ti for 1000 epoches. Different train time for 1000 epoches in various train strides. Batch size and train stride is fixed at 32 and 10.

4.3. Comparision with Other Methods

In this section, we will compare the training performance of different methods on this dataset. This includes a vanilla baseline based on CNN and GRU[5], as well as the state-of-the-art method based on multi-layer LSTM proposed in [4]. From Tab. 1, we can observe that our method exhibits superior performance than existing methods, achieves the 9.0%/1.5%/0.1% improvement in p_5, p_{10}, and p_{15}, and reduces the error distance to 2 pixels. Our network achieves excellent performance while effectively controlling the scale of parameters and computational operations. As shown in the Tab. 2, we need shorter training time compared to other methods.

4.4. Ablation Study

In this section, we will investigate the impact of different factors on the results in detail, including settings of the GRU

GRU		LTV-SSM	Metrics		
Uni	Bi		p_5	p_{10}	p_{error}
✓	✗	✗	0.916	**0.981**	**2.27**
✓	✗	✓	0.901	0.972	2.68
✗	✓	✗	<u>0.919</u>	**0.981**	<u>2.29</u>
✗	✓	✓	**0.935**	<u>0.976</u>	2.35

Table 3. Comparison of different structure designs in the dual recurrent module. We adapted bi-directional and uni- version of GRU, and attempted to remove LTV-SSM module in each instance.

Event Representation	p_5	p_{10}	p_{error}
Voxel-grid	0.891	<u>0.978</u>	2.56
Frame	<u>0.912</u>	**0.990**	<u>2.42</u>
Bina-rep	**0.935**	0.976	**2.35**

Table 4. Comparison on the different event representations.

and the LTV-SSM in the Dual Recurrent Module, Event representations, and data augmentation techniques.

Bi-GRU & LTV-SSM in Dual Recurrent Module: To verify the effectiveness of the Dual Recurrent Module design, we attempt to remove the Bi-GRU or LTV-SSM in it and observe how the performance changes. The results are shown in Tab. 3, which demonstrates the validness of the Dual Recurrent Module. Specifically, adopting Uni-GRU to replace the Bi-GRU causes the significant tracking error increase in p_{error} and remarkable missing rate increase in p_5 and p_{10}. It is owing to the lack of context information at the start and end positions of each sequence and bidirectional information helps to capture the valid pupil state to cope the difficult scenarios. In further, removing the LTV-SSM results in an increase in the prediction error by nearly 0.1 pixel, and affects the stability of the predictions by doubling the missing rate within 5 and 10 pixels. It dues to the LTV-SSM exploit the contextual relationship selectively, which helps to cast more attention in informative timestamp, like the smooth pursuit phase and less attention in resting or blinking phase. Thus, it can contribute to the stability of prediction and especially brings the improvement on p_5 metric.

Event Representations: This part evaluates the impact of different event representations. From Tab. 4, it can be seen that the Bina-rep helps to improve the tracking accuracy effectively, especially within a 5-pixel range. It indicates that the Bina-rep contributes to a more stable tracking curve by reducing the influence of other interference factors, e.g., time jitters from the camera. Thus, it can improve the stability of the model, even in the fast eye movement phase.

Event-Cutout and other data enhancing techniques:

Augmentations	p_5	p_{10}	p_{error}
w/o spatial shift	0.872	0.956	3.37
w/o temporal shift	<u>0.906</u>	<u>0.974</u>	<u>2.51</u>
w/o spatial flip	0.888	0.961	2.90
w/o Event-Cutout	0.895	0.959	2.83
Full Augmentations	**0.935**	**0.976**	**2.35**

Table 5. Comparison on the different data augmentation techniques.

The data enhancement techniques contribute to improving the accuracy of predictions under challenging scenarios, e.g., eye blinking. Adopted augmentation techniques include: 1) Position offsets: It is usually hard to estimate when pupils' position is on the edge of the frame. Shifting spatial positions of samples helps to obtain more edge samples for training, thus enhancing performance in such conditions; 2) Spatial flips: randomly flipping the frame, especially on the Y-axis, helps to understand the motion pattern of eyes in different viewing angles. As it is rare to have eye frames upside down to train the network, such a flip can enhance the generalization of the model; 3) Time shift: We slice the raw event using a fixed time window, and it carries the risk of information loss, which can be alleviated by utilizing a time shift. We conducted ablation studies on both the event cutout method and the above data augmentation techniques; the results are shown in Tab. 5. It can be seen that these data enhancing techniques prove more difficult samples to improve the model's generalization ability, especially the spatial transformation, *e.g.*, spatial shift and event-cutout, create the most valid adversarial samples.

4.5. Qualitative Comparison

In this section, we conduct qualitative comparisons of different models and structure designs in dual recurrent modules to comprehensively verify the effectiveness of the MambaPupil. The visualization results are shown in Fig. 5 and Fig. 6. We selected four typical and also challenging scenarios: **Onset of event slice**: At the initial end of a segment, there is typically a lack of contextual information, which poses a significant disadvantage for RNN networks relying on temporal correlations; **Blinking and rapid eye movement process**: Blinking generates a large number of interfering event, which can cause the network's prediction results to shift and jitter; **Eye rest**: Event cameras struggle to capture static objects, resulting in minimal information available to infer pupil position when the eyes are still. These scenarios can cause poor performance in predictions of existing methods. However, the visual results proved that the MambaPupil overcomes these challenges and still achieves high as well as stable tracking accuracy compared to the CNN-GRU and CB-ConvLSTM model. To delve

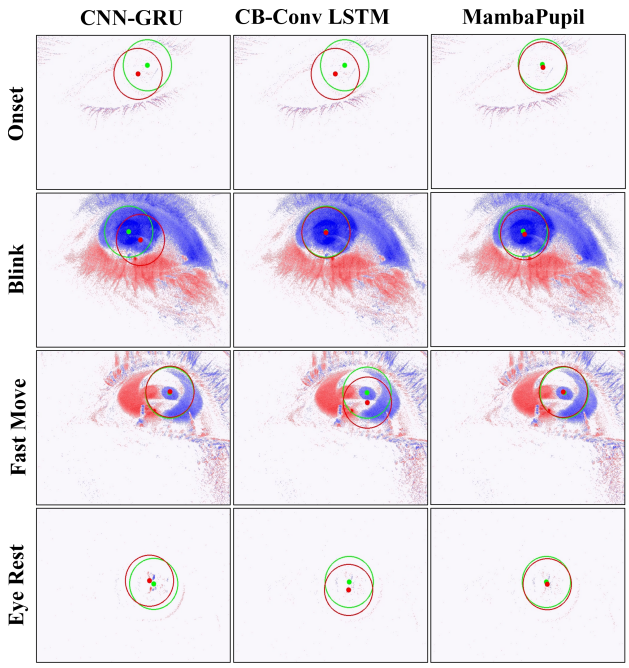

Figure 5. Visual comparison of state-of-the-art model and our MambaPupil in four challenging scenarios, including onset, blink, fast move and eye rest.

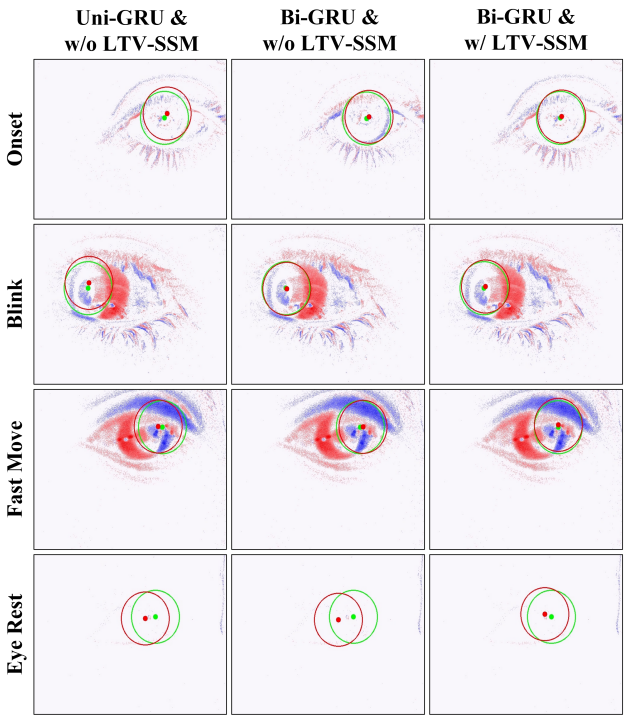

Figure 6. Qualitative comparison of structure design in dual recurrent module.

deeper into the roles played by different modules in the network, we sequentially discarded the LTV-SSM module and bidirectional design in Bi-GRU and observed a remarkable performance loss in prediction in Fig. 6.

5. Conclusion

In this paper, we analyzed the eye tracking task based on pure event data and proposed MambaPupil framework that integrates Bi-GRU and LTV-SSM modules to fully obtain temporal correlations, enabling accurate and fast-tracking of pupil position. Furthermore, with the Bina-reps and delicately designed data enhancement, we outperform the state-of-the-art model and achieve excellent results even in challenging scenarios. We believe this approach can provide valuable insights for other motion-tracking tasks. For future work, we aim to fully leverage the asynchronous nature of event data and further explore the generalization potential of state space models in other event-based tasks.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (NSFC) under Grants 62225207 and 62306295.

References

[1] Sami Barchid, José Mennesson, and Chaabane Djeraba. Bina-rep event frames: A simple and effective representation for event-based cameras. In *IEEE International Conference on Image Processing*, pages 3998–4002, 2022. 2, 4

[2] Pietro Bonazzi, Sizhen Bian, Giovanni Lippolis, Yawei Li, Sadique Sheik, and Michele Magno. A low-power neuromorphic approach for efficient eye-tracking. *arXiv preprint arXiv:2312.00425*, 2023. 1, 3

[3] Pietro Bonazzi, Thomas Rüegg, Sizhen Bian, Yawei Li, and Michele Magno. Tinytracker: Ultra-fast and ultra-low-power edge vision in-sensor for gaze estimation. *IEEE Sensors*, pages 1–4, 2023. 3

[4] Qinyu Chen, Zuowen Wang, Shih-Chii Liu, and Chang Gao. 3et: Efficient event-based eye tracking using a change-based convlstm network. In *IEEE Biomedical Circuits and Systems Conference*, pages 1–5, 2023. 1, 3, 6

[5] ChrisJudy, Nanashi, and Zuowen Wang. Event-based eye tracking - ais2024 cvpr workshop, 2024. https://kaggle.com/competitions/event-based-eye-tracking-ais2024. 6

[6] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Conference on Neural Information Processing Systems Workshop on Deep Learning*, 2014. 5

[7] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 44(01):154–180, 2022. 3

[8] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 2, 3, 5

[9] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *The International Conference on Learning Representations*, 2022. 3

[10] Benedikt Hosp, Shahram Eivazi, Maximilian Maurer, Wolfgang Fuhl, David Geisler, and Enkelejda Kasneci. Remoteeye: An open-source high-speed remote eye tracker: Implementation insights of a pupil- and glint-detection algorithm for high-speed remote eye tracking. *Behavior Research Methods*, 52:1387–1401, 2020. 2

[11] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *International Conference on Computer Vision*, page 6912–6921, 2019. 3

[12] Simon Klenk, David Bonello, Lukas Koestler, Nikita Araslanov, and Daniel Cremers. Masked event modeling: Self-supervised pretraining for event cameras. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024. 2, 4

[13] Kang Il Lee, Jung Ho Jeon, and Byung Cheol Song. Deep learning-based pupil center detection for fast and accurate eye tracking system. In *European Conference on Computer Vision*, pages 36–52, 2020. 2

[14] Joseph Lemley, Anuradha Kar, and Peter Corcoran. Eye tracking in augmented spaces: A deep learning approach. In *2018 IEEE Games, Entertainment, Media Conference*, pages 1–6, 2018. 1

[15] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, pages 24–26, 2017. 6

[16] Maria Laura Mele and Stefano Federici. Gaze and eye-tracking solutions for psychological research. 13:261–265, 2012. 1

[17] Clara Mestre, Josselin Gautier, and Jaume Pujol. Robust eye tracking based on multiple corneal reflections for clinical applications. *Journal of biomedical optics*, page 23, 2018. 2

[18] Cristina Palmero, Javier Selva, Mohammad Ali Bagheri, and Sergio Escalera. Recurrent CNN for 3d gaze estimation using appearance and shape cues. In *British Machine Vision Conference*, page 251, 2018. 3

[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. 32: 8024–8035, 2019. 6

[20] Julian Steil, Michael Xuelin Huang, and A. Bulling. Fixation detection for head-mounted eye tracking based on visual similarity of gaze targets. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, pages 23:1–23:9, 2018. 1

[21] T. Stoffregen, H. Daraei, C. Robinson, and A. Fix. Event-based kilohertz eye tracking using coded differential lighting. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3937–3945, 2022. 3

[22] Engin Türetkin, Sareh Saeedi, Siavash Bigdeli, Patrick Stadelmann, Nicolas Cantale, Luis Lutnyk, Martin Raubal, and Andrea L Dunbar. Real time eye gaze tracking for human machine interaction in the cockpit. *AI and Optical Data Sciences III*, pages 24–33, 2022. 2

[23] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 5

[24] 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. 6

[25] Wei Zhai, Yang Cao, Jing Zhang, and Zheng-Jun Zha. Exploring figure-ground assignment mechanism in perceptual organization. *Advances in Neural Information Processing Systems*, 35:17030–17042, 2022. 2

[26] Wei Zhai, Hongchen Luo, Jing Zhang, Yang Cao, and Dacheng Tao. One-shot object affordance detection in the wild. *International Journal of Computer Vision*, 130(10): 2472–2500, 2022. 3

[27] Wei Zhai, Yang Cao, Jing Zhang, Haiyong Xie, Dacheng Tao, and Zheng-Jun Zha. On exploring multiplicity of primitives and attributes for texture recognition in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 3

[28] Wei Zhai, Pingyu Wu, Kai Zhu, Yang Cao, Feng Wu, and Zheng-Jun Zha. Background activation suppression for weakly supervised object localization and semantic segmentation. *International Journal of Computer Vision*, 132(3): 750–775, 2024. 2

[29] Xucong Zhang, Yusuke Sugano, and Andreas Bulling. Evaluation of appearance-based methods and implications for gaze-based applications. pages 1–13, 2019. 3

[30] Nikola Zubić, Mathias Gehrig, and Davide Scaramuzza. State space models for event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3