RTDNet: A Recurrent Network using Temporal Difference for Video Gaze Estimation

Anonymous WACV Algorithms Track submission

Paper ID 1868

Abstract

Accurate gaze prediction is crucial in various fields. Recently, predicting gaze direction from videos has gained increasing attention, but traditional methods mainly rely on static images, making it challenging to leverage temporal and motion information. To address this, we propose RTDNet, a Recurrent Network using Temporal Difference for video gaze estimation, which integrates inter-frame motion and temporal sequence information to improve accuracy. First, we introduce the Temporal Difference Module (TDModule) to capture inter-frame motion information for better gaze estimation. It extracts motion information through temporal differencing and grouped convolution. We then upsample these features to create a more complete motion representation, combining them with original video features at different levels through a recurrent structure. We also employ Transformer to enhance the processing of temporal sequences. Additionally, we introduce Point Distribution Alignment Loss (PDA Loss), which aligns Points of Gaze (PoG) trajectories by using sequence information, further improving accuracy. Our approach achieves state-of-the-art performance on the EVE and EYEDIAP datasets.

1. Introduction

Human gaze conveys rich intent and plays a crucial role in fields such as VR/AR [33, 38], human-computer interaction [13, 20], and security monitoring [22, 34]. This process involves the transmission of substantial psychological and action-related information [27], making the study of accurate gaze estimation methods highly valuable.

In computer vision domain, learning-based gaze estimation methods primarily infer human gaze direction from appearance images [10, 25]. These approaches use webcams to capture facial or eye images and learn the mapping between appearance and gaze direction [8, 11, 24, 31, 39, 40]. While popular for their low cost and nice performance [10], they are mainly limited to static image analysis and do not fully account for temporal dynamics in video data.

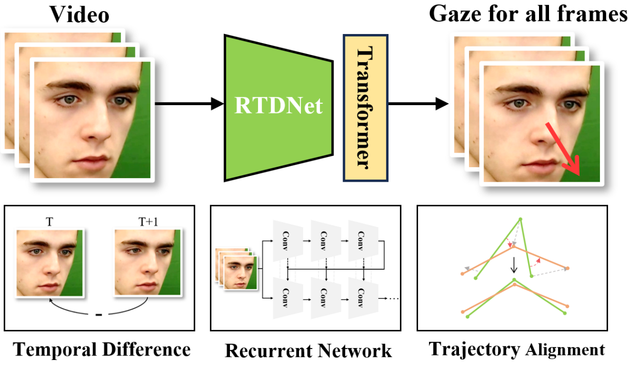


Figure 1. **Overview of our method.** We propose the RTDNet for video gaze estimation. It extracts temporal difference features between frames and employs a recurrent network to integrate these features at different levels, then utilizing a Transformer to leverage temporal information further. We also proposed the PDA Loss to better align the predicted gaze trajectories with the ground truth.

In contrast, videos, consisting of a sequence of continuous frames, provide valuable supplementary data for gaze prediction through inter-frame differences that reflect motion information [2, 3, 37]. The motion information describes the movement of the face, providing subtle cues that enrich the context for more accurate dynamic gaze estimation. [21]. Moreover, as human gaze follows a continuous trajectory, the gaze direction in surrounding frames provides valuable context for estimating the gaze in the current frame. Combining temporal dynamics and motion information from videos with existing static image methods could positively impact gaze prediction [10].

Recent video gaze estimation methods often use CNN architectures to extract features from each frame and employ sequence-processing modules like LSTM [19] to integrate features across multiple frames, mapping the features to gaze direction [23, 30, 42]. To fully leverage inter-frame motion information and PoG sequence distribution information, we propose a Recurrent Network using Temporal Difference for video gaze estimation (RTDNet). We first define a Temporal Difference Module (TDModule) to extract inter-frame motion features. This module captures motion features by applying temporal differencing to video features combined with grouped convolution. Next, we define a recurrent network structure to integrate motion information and video features at different levels. Specifically, the deep motion features are upsampled and combined with shallow motion features to form a more complete motion representation. We then re-extract the video features and append the complete motion features to them. Additionally, we compared various sequence processing modules and selected the Transformer [2] as our method to further improve the accuracy and effectiveness of gaze prediction.

Moreover, traditional methods' loss functions typically evaluate estimation accuracy on static images. For gaze estimation in videos, most methods simply aggregate the loss from each frame without considering the temporal dynamics in the video [21, 23, 30, 42]. To address this, we propose Point Distribution Alignment Loss (PDA Loss), which leverages sequence information to compute the differences between gaze point trajectories, enhancing the similarity between the predicted and actual gaze trajectories, thereby improving gaze prediction accuracy.

Our main contributions are summarized as follows:

- We proposed a Temporal Difference Module (TDModule) that extracts inter-frame motion features from temporal difference video features. This module captures motion information in videos, improving the utilization of motion information in gaze estimation.
- We designed a Recurrent Network using Temporal Difference for video gaze estimation (RTDNet), it incorporates motion information into video features at different levels by utilizing the TDModule and a recurrent structure, and employs a Transformer to process features along the temporal dimension, ultimately mapping them to the gaze direction for each frame.
- We proposed Point Distribution Alignment Loss (PDA Loss), which aligns the PoG sequences from a distributional perspective, enhancing the accuracy of video gaze estimation and making the predicted gaze trajectory shapes more similar to the ground truth trajectories. Our model achieves state-of-the-art performance on the EVE and EYEDIAP datasets.

2. Related Work

Traditional gaze estimation methods often rely on constructing and fitting a 3D geometric eye model [1, 16, 17, 26, 43]. However, due to calibration needs, environmental dependence, and high equipment costs, research has shifted towards appearance-based techniques [10]. These methods can be categorized into image-based and video-based approaches based on the type of input data.

2.1. Image-based Gaze Estimation

Image-based gaze estimation methods use pictures as input. Typically, Convolutional Neural Networks (CNNs) are employed as backbone. Zhang et al. [39] first used CNNs to extract features from grayscale monocular images and map them to gaze vectors with head pose. Later, Zhang et al. [41] chose VGG16 as backbone, increased the prediction accuracy further. Chen et al. [6] utilized dilated CNNs to capture subtle eye variations. Cheng et al. [9] used a four-stream CNN with asymmetric regression to estimate gaze from binocular images. Some studies use face images for gaze estimation. Zhang et al. [40] introduced a spatial weighting mechanism to reduce facial feature noise. Cheng et al. [7] integrated Vision Transformers (ViT) with CNNs to improve accuracy. Balim et al. [4] proposed a U-Net-like architecture to predict the six-dimensional gaze ray directly from the camera frame, bypassing complex preprocessing. These studies suggest that face images yield better predictions than eye images alone, which is why we also use face as input of RTDNet in our research.

2.2. Video-based Gaze Estimation

Videos consist of consecutive frames, and dynamic methods often leverage temporal information to improve gaze estimation accuracy. Compared to static images, the temporal information in videos can enrich the context of gaze prediction and improve prediction accuracy [21, 30, 42]. Palmero et al. [29] proposed a recurrent CNN architecture that incorporates a recurrent module after feature extraction, effectively capturing temporal dependencies between frames and generating more accurate gaze direction predictions. Kellnhofer et al. [23] proposed a Gaze360 model employed a pinball bidirectional LSTM to process CNN outputs, taking into account not only the current frame but also past and future frames to refine the gaze estimation for the current frame, significantly improving prediction robustness. In addition, some studies have introduced dedicated datasets for video-based gaze estimation to advance the field. The EYEDIAP dataset [14] use screens or floating objects as targets, offering a rich variety of gaze targets and dynamic background information, making it suitable for validating video-based gaze estimation algorithms. The EVE dataset [30] is a large-scale video gaze dataset focused on screen-based gaze targets, where researchers processed image features using GRUs and refined initial predictions by incorporating heatmaps of screen content, leading to significantly improved prediction accuracy. Previous methods for handling temporal information inspired us to leverage the motion information contained in videos to improve gaze estimation accuracy. Our approach extracting motion features in video and integrating motion information with video features at different levels, explicitly leverages motion information to improve prediction accuracy.

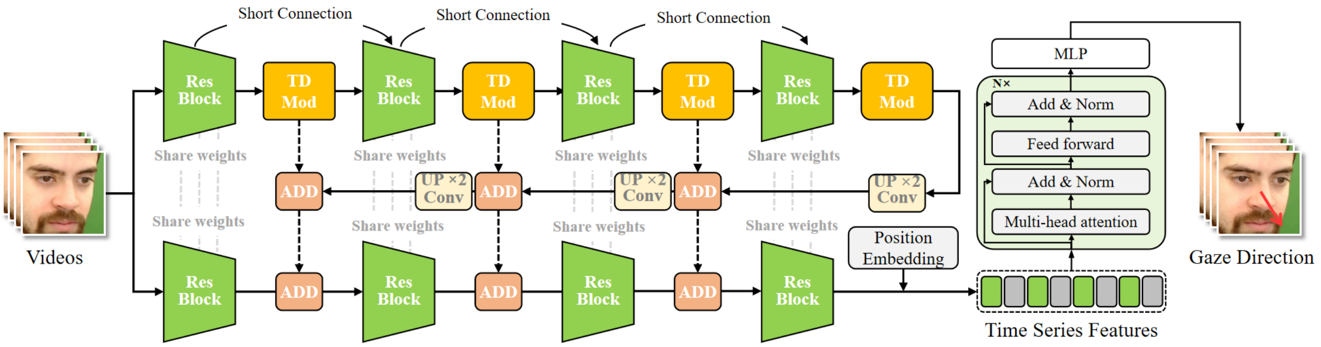


Figure 2. **Architecture of RTDNet.** RTDNet takes facial videos as input, extracts video features using ResBlock, and employs the TDModule to capture inter-frame motion information, which is then appended to the video features. After the fourth layer, the deep motion features are upsampled, convolved, and combined with shallow motion information to obtain a complete motion representation. This complete motion information is then integrated into the video features through a recurrent design and fed into a Transformer to process the features along the temporal dimension. Finally, the gaze direction for each frame is output through an MLP mapping.

3. Methodology

3.1. Overview

We propose a novel network RTDNet to predict gaze in videos by learning the inter-frame motion differences and aligning the gaze distribution. First, we would like to define the task and the notation used in the following sections. Our objective is to learn a mapping function $f : V \to G$, where $V \in \mathbb{R}^{T\times 3\times H\times W}$ represents the video input. Here, T denotes the video length, also known as the temporal dimension, and H and W denote the height and width of each frame, respectively. The output $G \in \mathbb{R}^{T\times 2}$ represents the gaze direction for all frames, expressed in terms of pitch and yaw angles. In this section, we will first introduce TDModule, then RTDNet, and finally PDA Loss.

3.2. Network Architecture

RTDNet is a recurrent network using temporal difference (TD) for video gaze estimation. The RTDNet inputs with video and outputs gaze directions for all frames. It contains a TDModule to capture motion feature and a recurrent network to incorporate high-level feature with low-level feature for gaze estimation.

3.2.1 Temporal Difference Module

We designed a TDModule to capture motion features from video. Overall, TDModule improves gaze estimation accuracy by extracting motion information from temporal difference features through grouped convolution.

Specifically, the TDModule takes video features as input, denoted as the set $F = \{f_t \mid t = 0, 1, \ldots, T-1\}$. To capture motion difference information between frames, we perform a temporal differencing operation on the extracted video feature maps, resulting in the difference feature set $\Delta F = \{\Delta f_t \mid \Delta f_t = f_{t+1} - f_t, t = 0, 1, \ldots, T-2\}$.

Next, our goal is to aggregate and group multiple difference features using a sliding window module, followed by applying a convolutional layer to each group to extract motion information. To ensure the number of motion features matches the number of frames, we perform zero-padding on the difference feature set. Specifically, we pad m zeros at the beginning and n zeros at the end of the sequence ΔF, ensuring that the final number of groups equals T. The zero-padded sliding window aggregation is formally defined as:

$$\widetilde{\Delta F} = \{\underbrace{0, 0, \ldots, 0}_{m\text{ zeros}}, \Delta f_0, \Delta f_1, \ldots, \Delta f_{T-2}, \underbrace{0, 0, \ldots, 0}_{n\text{ zeros}}\}, \tag{1}$$

$$\Delta F_{\text{win}}(w, s) = \{\widetilde{\Delta F}_i^{(w)} \mid i = 0, 0+s, 0+2s, \ldots\}. \tag{2}$$

Here, $\widetilde{\Delta F}_i^{(w)}$ represents the w consecutive elements starting from index i in the sequence $\widetilde{\Delta F}$, where $i + w - 1 <$ len$(\widetilde{\Delta F})$. s is the stride of the sliding window.

After applying sliding window aggregation, we apply convolutional layers to each group to extract motion information. The grouping operation also reduces the number of channels in the difference features, thereby decreasing the number of parameters in the convolutional layers. Subsequently, we add the extracted motion features to the original video features, resulting in video features enriched with motion information, which can be denoted as:

$$f_t^{'} = \text{Conv}\left(\widetilde{\Delta F}_i^{(w)}\right) + f_t. \tag{3}$$

We summarize the process of extracting motion features in this manner as the TDModule. We illustrate the basic process of the TDModule in Figure 3.

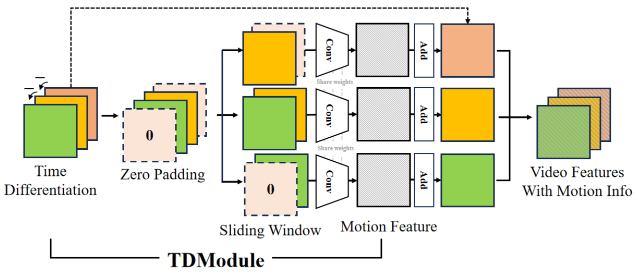


Figure 3. **TDModule and Short Connection.** First, the temporal difference video features are calculated, followed by zero-padding to ensure the number of convolutional feature maps matches the number of video frames. Next, the difference features are aggregated using a sliding window, processed through convolution, and finally added to the original video features.

3.2.2 Recurrent Network Design

We design a recurrent network to estimate gaze directions from video, as shown in Figure 2. In general, this network effectively integrates the motion features extracted by the TDModule with video features at different levels, improving gaze estimation accuracy. Initially, we fuse video features and motion information through short connections. Deep motion features are then upsampled and combined with shallow features to form a complete motion representation, which is integrated with video features at different levels through the recurrent structure. Finally, a Transformer leverages the temporal information, and a multilayer perceptron maps the features to gaze directions.

Specifically, the network uses video as input. First, we use a ResBlock to extract the features of each frame in the video. The ResBlock is the basic residual block from ResNet [18] which employs residual connections. Then, we apply the TDModule, as described in the previous section, to extract motion information from the video features. We then add it to the original video features, thereby embedding the motion information into the features of each frame. This process is then repeated for deeper motion features.

In the fourth layer, after the TDModule extracts motion features, we omit the short connection. As the network deepens, the TDModule captures more global motion information, but the reduced feature map size limits the capture of fine motion details. In contrast, shallow motion feature maps retain detailed motion information but lack global context. This trade-off results in each layer's motion features having certain limitations. To balance global and detailed motion information at each layer, inspired by the Hourglass Network [28], we upsample the deeper feature maps to enlarge their spatial dimensions and apply convolution to reduce channels. This reshapes the deep motion features to match the dimensions of the previous layer's features. These upsampled features are added to the previous layer's incomplete motion features, compensating for a complete motion representation. This process is repeated to progressively enhance the motion features at each layer.

Then, we use the same ResBlock to re-extract the video features and append the complete motion features to them. After repeating this process four times, we obtain video features enriched with extensive motion information.

Next, considering that human gaze shifts continuously, the gaze directions in consecutive frames of a video serve as an important reference for predicting the gaze direction of the current frame. Therefore, it is necessary to introduce a module to handle the temporal variations in features. We chose to use the Transformer Encoder [36] for this purpose. The Transformer, based on the self-attention mechanism, has been widely proven effective in the field of nature language processing [32]. Its multi-head self-attention mechanism can simultaneously compute attention in different subspaces, thereby capturing complex dependencies between different time steps in a sequence, which is beneficial for gaze estimation in videos. The features processed by the Transformer are then passed into a multilayer perceptron (MLP), where they are finally mapped to the gaze directions in the form of pitch and yaw. The entire RTDNet process is written in Algorithm 1. We also considered traditional temporal processing modules such as RNN [35] and its variants, GRU [12] and LSTM [19]. A detailed comparison between these approaches is discussed in Section 4.6.

3.3. Point Distribution Alignment Loss

The loss function consists of two parts, our proposed Points Distribution Alignment (PDA) Loss and Angular Loss. Angular Loss is calculated from the 3D gaze vector, and PDA Loss is calculated from the PoG sequence. The whole loss function $\mathcal{L}$ can be denoted as:

$$\mathcal{L} = \mathcal{L}_{angular} + \gamma \mathcal{L}_{PDA}. \tag{4}$$

where γ is a coefficient to balance these two losses.

PDA Loss. We obtain the PoG by calculating the intersection of the gaze vector with the screen. In the given context, $P = (p_0, \ldots, p_{T-1})$ represents the predicted sequence of PoGs, and $Q = (q_0, \ldots, q_{T-1})$ represents the corresponding ground truth. Subsequently, we calculated the directional vector $\vec{v}$ between each point. Specifically, $\vec{v}_i^P = p_{i+1} - p_i$ and $\vec{v}_i^Q = q_{i+1} - q_i$.

The directional vector $\vec{v}$ measures the direction of the gaze trajectory. To utilize the sequence information of the PoG, we calculate the angular difference θ between each trunk of trajectory. Here, θ_i represents the angular difference for the ground truth trajectory, and $\hat{\theta}_i$ represents the angular difference for the predicted trajectory. The calculation formula for $\hat{\theta}$ is provided as follows:

$$\hat{\theta}_i = \arccos\left(\frac{\vec{v}_i^P \cdot \vec{v}_{i+1}^P}{\|\vec{v}_i^P\|\|\vec{v}_{i+1}^P\|)}\right), \tag{5}$$

Algorithm 1 RTDNet

```
Require: Video $V$
Ensure: Gaze Direction $G$ For All Frames
Motion Feature Extraction:
$F_v^0 \leftarrow V$
for $i = 1$ to 3 do
    $F_{res}^i \leftarrow ResBlock(F_v^{i-1})$
    $F_{motion}^i \leftarrow TDModule(F_{res}^i)$
    $F_v^i \leftarrow F_{res}^i + F_{motion}^i$
end for
Deep + Shallow Motion Fusion:
$F_{motion\_full}^4 \leftarrow TDModule(ResBlock(F_v^3))$
for $i = 3$ to 1 do
    $F_{up\_motion}^{i+1} \leftarrow \text{Upsample}(F_{motion\_full}^{i+1})$
    $F_{motion\_full}^i \leftarrow \text{Conv}(F_{up\_motion}^{i+1}) + F_{motion}^i$
end for
Video and Motion Feature Fusion:
{ResBlock shares weights with earlier ResBlock}
$F_{comb}^0 \leftarrow V$
for $i = 1$ to 3 do
    $F_{comb}^i \leftarrow ResBlock(F_{comb}^{i-1}) + F_{motion\_full}^i$
end for
$F_{final}^4 \leftarrow ResBlock(F_{comb}^3)$
Temporal Process and Map to Gaze:
$F_{pos} \leftarrow \text{Position Embedding}(F_{final}^4)$
$G \leftarrow \text{MLP}(\text{Transformer}(F_{pos}))$
return $G$
```

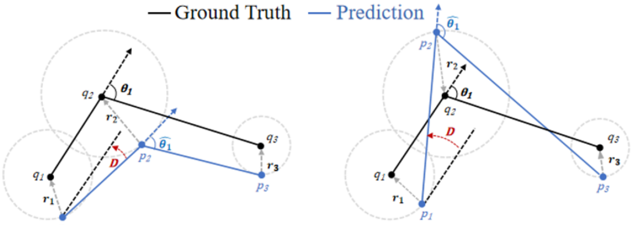
After calculating the angular differences within the PoG sequences, we define a new metric D to measure the angular differences between the trajectories of two PoG sequences:

$$D = \sum_{i=0}^{T-3} |\theta_i - \hat{\theta}_i|, \tag{6}$$

A reduction in D makes the two PoG trajectories more parallel, but relying solely on D may result in parallel but non-overlapping trajectories. To avoid this, we include the Euclidean distance to ensure each PoG aligns with the ground truth. Thus, we define PDA Loss as follows:

$$\mathcal{L}_{PDA}(P, Q) = \sum_{i=0}^{T-1} \|p_i - q_i\| + \lambda D. \tag{7}$$

where $\lambda \in \mathbb{R}_{>0}$ is a balance coefficient. Overall, PDA Loss minimizes the Euclidean distance between corresponding points of the two sequences while constraining their relative positional relationships through metric D. The working principle of PDA Loss is illustrated in Figure 4.



(a) PDA Loss with Low Angular Difference (D) (High Similarity)

(b) PDA Loss with High Angular Difference (D) (Low Similarity)

Figure 4. **Working principle of PDA Loss.** The points represent PoG, with the connecting lines illustrating gaze trajectories. As shown, the Euclidean distance part in PDA Loss makes r_i smaller, bringing each predicted PoG closer to the ground truth. The angular difference D measures the discrepancy between θ_i and $\hat{\theta}_i$, indicating the parallelism of the trajectories. While both (a) and (b) have the same Euclidean distance, (a)'s lower D value results in a predicted trajectory shape closer to the ground truth compared to (b). This demonstrates how PDA Loss aligns predicted trajectories by minimizing both distance and angular deviation.

Angular Loss. Let T denote the length of the video frame sequence. Convert both the predicted 3D gaze direction $\hat{\mathbf{g}}$ and the actual gaze direction $\mathbf{g}$ into their three-dimensional forms. The Angular Loss is then defined as:

$$\mathcal{L}_{\text{angular}}(\mathbf{g}, \hat{\mathbf{g}}) = \frac{1}{T} \sum^{T} \frac{180}{\pi} \arccos\left(\frac{\mathbf{g} \cdot \hat{\mathbf{g}}}{\|\mathbf{g}\| \|\hat{\mathbf{g}}\|}\right). \tag{8}$$

4. Experiments

4.1. Datasets

EVE [30] is an extensive video gaze dataset comprising over 12 million frames, collected from a total of 54 participants in a controlled laboratory setting. The dataset provides four synchronized camera views from different angles and includes a wide variety of visual stimuli, ultimately resulting in approximately 105 hours of recorded video data. The standard dataset split consists of 39 subjects in the training set, 5 subjects in the validation set, and 10 subjects in the test set. However, due to the unavailability of ground truth labels in the test set, we conduct evaluations of all models on the validation set for consistency.

EYEDIAP [14] is a highly adaptable gaze estimation dataset, specifically designed for the evaluation of algorithms using both RGB and RGB-D data. The dataset consists of recordings from 16 participants across a total of 94 sessions, capturing a wide range of variations in head pose, visual targets, and ambient conditions. It encompasses both static and dynamic head poses and features three distinct types of visual targets: discrete screen targets, continuous screen targets, and a 3D floating ball. Continuous screen targets are employed for model training and evaluation purposes. Given the absence of a default dataset split, we allocate 20% of the data as a test set. As EYEDIAP dataset lacks official facial videos, we followed the preprocessing and normalization method by Cheng et al. [10].

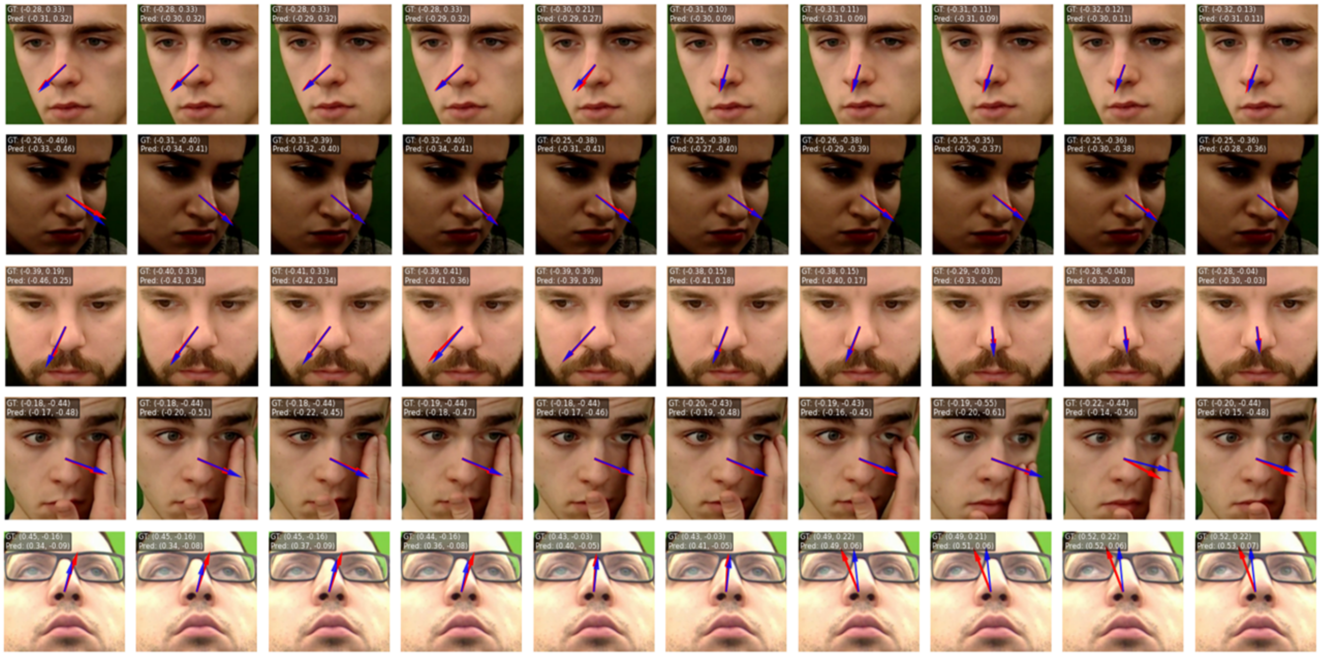

GT: (-0.28, 0.33) Pred: (-0.31, 0.32)
GT: (-0.28, 0.33) Pred: (-0.30, 0.32)
GT: (-0.28, 0.33) Pred: (-0.29, 0.32)
GT: (-0.28, 0.33) Pred: (-0.29, 0.32)
GT: (-0.30, 0.21) Pred: (-0.29, 0.27)
GT: (-0.31, 0.10) Pred: (-0.30, 0.09)
GT: (-0.31, 0.11) Pred: (-0.31, 0.09)
GT: (-0.31, 0.11) Pred: (-0.31, 0.09)
GT: (-0.32, 0.12) Pred: (-0.30, 0.11)
GT: (-0.32, 0.13) Pred: (-0.31, 0.11)
GT: (-0.26, -0.46) Pred: (-0.33, -0.46)
GT: (-0.31, -0.40) Pred: (-0.34, -0.41)
GT: (-0.31, -0.39) Pred: (-0.32, -0.40)
GT: (-0.32, -0.40) Pred: (-0.34, -0.41)
GT: (-0.25, -0.38) Pred: (-0.31, -0.41)
GT: (-0.25, -0.38) Pred: (-0.27, -0.40)
GT: (-0.26, -0.38) Pred: (-0.29, -0.39)
GT: (-0.25, -0.35) Pred: (-0.29, -0.37)
GT: (-0.25, -0.36) Pred: (-0.30, -0.38)
GT: (-0.25, -0.36) Pred: (-0.28, -0.36)
GT: (-0.39, 0.19) Pred: (-0.46, 0.25)
GT: (-0.40, 0.33) Pred: (-0.43, 0.34)
GT: (-0.41, 0.33) Pred: (-0.42, 0.34)
GT: (-0.39, 0.41) Pred: (-0.41, 0.36)
GT: (-0.39, 0.39) Pred: (-0.39, 0.39)
GT: (-0.38, 0.15) Pred: (-0.41, 0.18)
GT: (-0.38, 0.15) Pred: (-0.40, 0.17)
GT: (-0.29, -0.03) Pred: (-0.33, -0.02)
GT: (-0.28, -0.04) Pred: (-0.30, -0.03)
GT: (-0.28, -0.04) Pred: (-0.30, -0.03)
GT: (-0.18, -0.44) Pred: (-0.17, -0.48)
GT: (-0.18, -0.44) Pred: (-0.20, -0.51)
GT: (-0.18, -0.44) Pred: (-0.22, -0.45)
GT: (-0.19, -0.44) Pred: (-0.18, -0.47)
GT: (-0.18, -0.44) Pred: (-0.17, -0.46)
GT: (-0.20, -0.43) Pred: (-0.19, -0.48)
GT: (-0.19, -0.43) Pred: (-0.16, -0.45)
GT: (-0.19, -0.55) Pred: (-0.20, -0.61)
GT: (-0.22, -0.44) Pred: (-0.14, -0.56)
GT: (-0.20, -0.44) Pred: (-0.15, -0.48)
GT: (0.45, -0.16) Pred: (0.34, -0.09)
GT: (0.45, -0.16) Pred: (0.34, -0.08)
GT: (0.45, -0.16) Pred: (0.37, -0.09)
GT: (0.44, -0.16) Pred: (0.36, -0.08)
GT: (0.43, -0.03) Pred: (0.40, -0.05)
GT: (0.43, -0.03) Pred: (0.41, -0.05)
GT: (0.49, 0.22) Pred: (0.49, 0.06)
GT: (0.49, 0.21) Pred: (0.51, 0.06)
GT: (0.52, 0.22) Pred: (0.52, 0.06)
GT: (0.52, 0.22) Pred: (0.53, 0.07)

Figure 5. **Results of Gaze direction visualization.** This figure presents the visualization of the model's performance. The blue arrows represent the **predicted gaze direction**, while the red arrows indicate the **ground truth**. The top left corner of each frame is labeled with the gaze's pitch and yaw angle. The arrows depict the projection of the gaze direction in the 2D image, so their lengths are not significant. Each row corresponds to a different video. The first two rows show tests under varying camera angles and lighting conditions, the third row illustrates a rapid saccade, the fouth row demonstrates the impact of a hand partially obscuring and deforming the face, and the fifth row displays the prediction results when the subject is wearing glasses.

4.2. Implementation Details

For both EVE and EYEDIAP datasets, the input video sequence V comprises 30 frames, where each frame is sampled by taking every third frame from the original video sequence. Each frame has dimensions of $3 \times 256 \times 256$. In RTDNet, we use ResNet-18 [18] residual block as video feature extractor. For sliding window module, w and s were set as 5 and 1. Regarding zero-padding, m and n were set as 2 and 3, which means two zero feature maps are padded at the beginning, while three are padded at the end. Nearest-neighbor interpolation is employed for the upsampling process. The positional embeddings are set to be learnable. In the Transformer, we adopt an 8-head multi-head attention mechanism. We only use a single Transformer encoder layer to process temporal information, with an input dimension of 512, a feedforward network dimension of 1024, a dropout rate of 0.1, and GeLU as the activation function. The model was trained using the Adam optimizer with a weight decay of 0.005. For the loss function, the hyperparameters were configured as $\gamma = 0.0005$ and $\lambda = 1$. The initial learning rate was set to 0.0005 for the EVE dataset and 0.0001 for the EYEDIAP dataset. On the EVE dataset, the model was trained for a total of 40,960 iterations, with the learning rate halved every 4,096 iterations, and a batch size of 16 was used. For the EYEDIAP dataset, training spanned 10,000 iterations, with the learning rate halved every 2,000 iterations, and a batch size of 8 was employed. The model was implemented using PyTorch 2.3.

4.3. Performance Comparison on Two Datasets

We selected the EyeNet-GRU variant [30] as the baseline model for comparison with our method. To ensure fairness, we substituted its input features with facial images, utilized ResNet-18 as the backbone. This baseline model is referred to as FaceNet-GRU in Table 1. Additionally, several other appearance-based gaze estimation methods [4, 7, 23, 40, 41] were also included for comparison. We enhanced some static methods with a two-layer, bidirectionally connected LSTM layer, making the network recurrent to leverage temporal information in video. This bi-LSTM module is used in Gaze360 [23] to process temporal information. All models are trained for 40,960 iterations on the EVE model and for 10,000 iterations on EYEDIAP.

The comparison results are presented in Table 1. Note that some methods [5, 30] achieve better performance through calibration or refinement of PoG prediction results, with their improvements not primarily stemming from the model itself. We include these methods in the table for reference. Compared to the baseline model, our model

Table 1. **Performance Comparison on EVE and EYEDIAP Datasets.** The results showed the average angular difference between the ground truth and the prediction. Methods marked with * are refined either through calibration of the initial prediction results or by incorporating screen content, which leads to an unfair comparison. We also list them in the table for reference.

Model	EVE(°)	EYEDIAP(°)
EFE [4]	3.53	-
FullFace [40]	4.73	6.28
FullFace+LSTM [40]	4.60	6.21
GazeNet [41]	6.56	6.80
GazeNet+LSTM [41]	5.14	6.59
GazeTR-Hybrid [7]	4.35	5.22
Gaze360 [23]	4.27	5.24
EyeNet-GRU [30]	4.42	6.31
FaceNet-GRU [30]	3.22	5.42
GazeRefineNet * [30]	2.49	-
EVE-SCPT * [5]	1.95	-
RTDNet	**2.64**	**5.21**

achieves a 18.0% reduction in error on the EVE dataset and a 4.0% reduction on the EYEDIAP dataset, reaching state-of-the-art performance. We noted that our method exhibited a greater performance lead on the EVE dataset compared to the EYEDIAP. This is likely due to the low-resolution facial images present in EYEDIAP [10]. We observed that the performance of the static model was enhanced after the integration of a bidirectional LSTM, further substantiating the significance of temporal information in video gaze estimation. This supports the results of previous video models [10]. Additionally, models using facial images as input consistently show lower error rates compared to those using eye images on average, which aligns with findings from previous studies. Since EFE [4] have not publicly released their code and its structures are difficult to reproduce, we only present the results from its report for reference.

4.4. Ablation Study

In this section, we conduct an ablation study. As shown in Table 2, We perform two ablation experiments focusing on the Recurrent architecture and PDA Loss . All models for ablation studies are trained and evaluated on the EVE dataset. We have two setups:

w/o Recurrent network. We use ResNet-18 as the video feature extractor, with Angular Loss and PDA Loss as the loss functions. RTDNet utilizes a recurrent network design to extract motion features and fuse them with video features at different levels, enhancing gaze prediction accuracy. It first extracts motion features through the TDModule, then upsampling them to obtain a complete motion representation, and fusing these complete motion features with original video features at different levels through a recurrent structure. To evaluate the recurrent design, we perform an ablation study by replacing the recurrent network with ResNet-18, directly connecting it to the Transformer.

Table 2. Ablation Study on recurrent architecture and PDA Loss

Setup	Angular Difference (°)
w/o Recurrent network	3.18
w/o PDA Loss	2.72
RTDNet with PDA Loss	**2.64**

w/o PDA Loss. PDA Loss aligns PoG distributions by leveraging PoG sequence information. It calculates angular differences between gaze trajectories, combined with Euclidean distance, to better align the two PoG trajectories. In this ablation study, we train RTDNet using only Angular Loss, omitting PDA Loss to assess its impact.

The results of ablation study are shown in the Table 2. Compared to the model using a recurrent network, the performance of the model relying solely on ResNet-18 for video feature extraction decreased by 20.5%, proving the effectiveness of our recurrent network design in utilizing motion information. Furthermore, the performance dropped by 3.0% after removing PDA Loss, compared to the model trained with PDA Loss. This clearly demonstrates the performance improvement brought by PDA Loss in aligning PoG trajectories with ground truth through distribution information, proving the effectiveness of PDA Loss.

4.5. Evaluation of Sequence Similarity

In this section, we quantitatively analyze the impact of PDA Loss on PoG sequence alignment. We compare the performance of models trained with and without PDA Loss in predicting gaze point sequences on the EVE dataset. Notably, although we previously mentioned that PoG is not calculated during inference due to the absence of the origin of gaze, for the sake of comparing the alignment effect of PDA Loss, we also input the gaze origin coordinates during inference. We use Procrustes distance [15] as a metric to evaluate the similarity between the two point sequences. Procrustes analysis involves translating, rotating, and scaling two shapes to align them as closely as possible, followed by calculating the difference between the transformed shapes, which is represented by the Procrustes distance. Table 3 illustrates the average Procrustes distance of PoG (in pixels) across EVE dataset. The results show that with PDA Loss enabled, the average Procrustes distance decreased by 5.2% compared to the model without PDA Loss, demonstrating the effectiveness of PDA Loss.

4.6. Temporal Module Replacement Experiment

In this section, we evaluated the performance of different Temporal Process Modules. All models were trained and tested on the EVE dataset. The RNN and its variants we

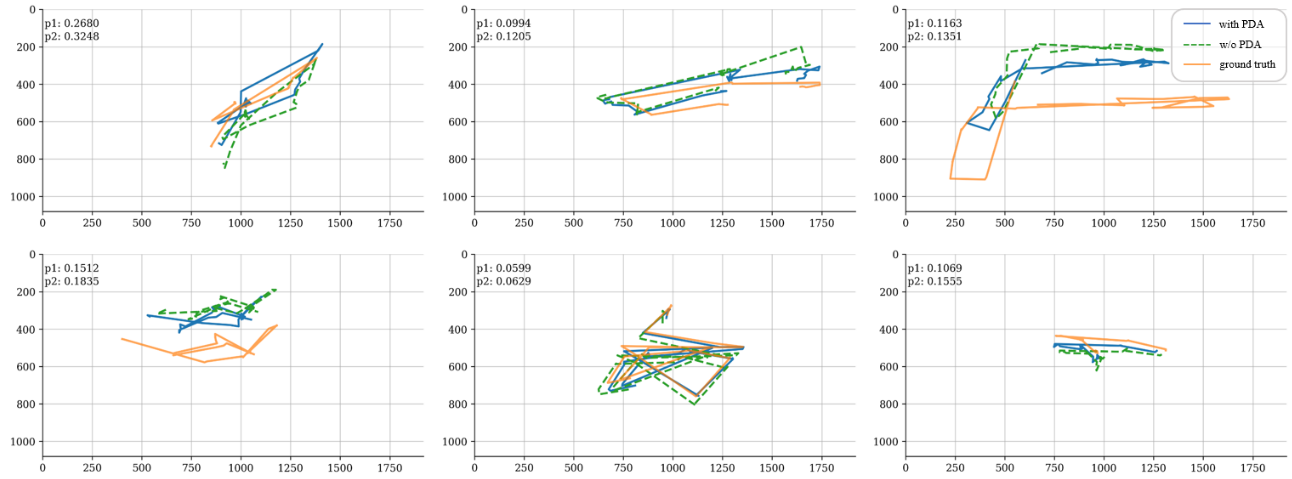


Figure 6. **Results of Visual Trajectory.**This figure depicts the gaze trajectory of a person on the screen, which is completed by connecting the points of gaze. The orange line in the image represents the **ground truth**, the blue line represents the prediction results of the **model using PDA Loss**, and the green color represents the model prediction results **without using the PDA Loss**. In the top left corner of each image, p1 and p2 respectively indicate the Procrustes distance with and without PDA Loss. The unit of the coordinates is the number of pixels, with the screen resolution being 1920 × 1080.

Table 3. **Evaluation of Sequence Similarity.** This table shows the effect of PDA Loss on PoG sequence alignment, measured by Procrustes distance. PDA Loss reduces the Procrustes distance, making the predicted and actual trajectories more similar in shape.

Model	**Procrustes distance [15]**
w/o PDA Loss	0.271
with PDA Loss	**0.257**

used here have both an input dimension and a hidden layer dimension of 512. As shown in Table 4, the results indicate that selecting the Transformer as our Temporal Process Module leads to the best performance.

4.7. Qualitative Evaluation

We conducted qualitative analyses, visualizing gaze predictions in Figure 5. The results show that our model accurately predicts gaze direction under varying camera angles and lighting. The third row demonstrates that RTDNet leverages motion information to assist gaze prediction during rapid saccades. In the fourth row, despite the subject's face being partially obscured by hand movements, the model maintained accuracy, highlighting RTDNet's robustness by leveraging motion and temporal information. However, prediction errors increased with subjects wearing glasses, likely due to limited training examples.

In Figure 6, we visualized the PoG trajectories inferred by models trained with and without PDA Loss. The trajectories with PDA Loss are noticeably closer to the ground truth in terms of both trajectory shape and Euclidean distance, clearly demonstrating its effectiveness compared to the trajectories generated without PDA Loss.

Table 4. **Performance of Different Temporal Process Modules on the EVE Dataset.** Among them, the LSTM and GRU variants demonstrated similar performance, both outperforming the RNN variant. The Transformer variant exhibited the best performance.

Model	**Angular Difference(°)**
RTDNet with RNN	3.08
RTDNet with LSTM	2.90
RTDNet with GRU	2.92
RTDNet with Transformer	**2.64**

5. Conclusion

In this paper, we address the limitations of previous video gaze estimation methods that did not fully leverage the motion and temporal information in videos. We propose a Recurrent Network using Temporal Difference for video gaze estimation (RTDNet), along with its key component, the Temporal Difference Module (TDModule), and the PDA Loss function. The TDModule extracts motion information from video features through temporal differencing and grouped convolutions. Subsequently, we create a more complete representation of motion features via upsampling and fuse these motion features with video features at different levels through a recurrent process. We use Transformer to handle temporal information. PDA Loss aligns PoG trajectories using video sequence information, improving prediction accuracy. RTDNet achieves state-of-the-art performance on both EVE and EYEDIAP datasets.

References

[1] Kenneth Alberto Funes Mora and Jean-Marc Odobez. Geometric generative gaze estimation (g3e) for remote rgb-d cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1773–1780, 2014. 2

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021. 1, 2

[3] Mihai Bace, Vincent Becker, Chenyang Wang, and Andreas Bulling. Combining gaze estimation and optical flow for pursuits interaction. In *ACM Symposium on Eye Tracking Research and Applications*, pages 1–10, 2020. 1

[4] Haldun Balim, Seonwook Park, Xi Wang, Xucong Zhang, and Otmar Hilliges. Efe: End-to-end frame-to-gaze estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2688–2697, 2023. 2, 6, 7

[5] Jun Bao, Buyu Liu, and Jun Yu. An individual-difference-aware model for cross-person gaze estimation. *IEEE Transactions on Image Processing*, 31:3322–3333, 2022. 6, 7

[6] Zhaokang Chen and Bertram E Shi. Appearance-based gaze estimation using dilated-convolutions. In *Asian Conference on Computer Vision*, pages 309–324. Springer, 2018. 2

[7] Yihua Cheng and Feng Lu. Gaze estimation using transformer. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 3341–3347. IEEE, 2022. 2, 6, 7

[8] Yihua Cheng, Feng Lu, and Xucong Zhang. Appearance-based gaze estimation via evaluation-guided asymmetric regression. In *Proceedings of the European conference on computer vision (ECCV)*, pages 100–115, 2018. 1

[9] Yihua Cheng, Feng Lu, and Xucong Zhang. Appearance-based gaze estimation via evaluation-guided asymmetric regression. In *Proceedings of the European conference on computer vision (ECCV)*, pages 100–115, 2018. 2

[10] Yihua Cheng, Haofei Wang, Yiwei Bao, and Feng Lu. Appearance-based gaze estimation with deep learning: A review and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1, 2, 5, 7

[11] Yihua Cheng, Xucong Zhang, Feng Lu, and Yoichi Sato. Gaze estimation by exploring two-eye asymmetry. *IEEE Transactions on Image Processing*, 29:5259–5272, 2020. 1

[12] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 4

[13] Heiko Drewes. *Eye gaze tracking for human computer interaction*. PhD thesis, lmu, 2010. 1

[14] Kenneth Alberto Funes Mora, Florent Monay, and Jean-Marc Odobez. Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras. In *Proceedings of the symposium on eye tracking research and applications*, pages 255–258, 2014. 2, 5

[15] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40:33–51, 1975. 7, 8

[16] Elias Daniel Guestrin and Moshe Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on biomedical engineering*, 53(6):1124–1133, 2006. 2

[17] Dan Witzner Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):478–500, 2009. 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 6

[19] S Hochreiter. Long short-term memory. *Neural Computation MIT-Press*, 1997. 1, 4

[20] Thomas E Hutchinson, K Preston White, Worthy N Martin, Kelly C Reichert, and Lisa A Frey. Human-computer interaction using eye-gaze input. *IEEE Transactions on systems, man, and cybernetics*, 19(6):1527–1534, 1989. 1

[21] Swati Jindal, Mohit Yadav, and Roberto Manduchi. Spatio-temporal attention and gaussian processes for personalized video gaze estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 604–614, 2024. 1, 2

[22] Christina Katsini, Yasmeen Abdrabou, George E Raptis, Mohamed Khamis, and Florian Alt. The role of eye gaze in security and privacy applications: Survey and future hci research directions. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–21, 2020. 1

[23] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6912–6921, 2019. 1, 2, 6, 7

[24] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2176–2184, 2016. 1

[25] Yaxiong Lei, Shijing He, Mohamed Khamis, and Juan Ye. An end-to-end review of gaze estimation and its interactive applications on handheld mobile devices. *ACM Computing Surveys*, 56(2):1–38, 2023. 1

[26] Feng Lu, Yue Gao, and Xiaowu Chen. Estimating 3d gaze directions using unlabeled eye images via synthetic iris appearance fitting. *IEEE Transactions on Multimedia*, 18(9):1772–1782, 2016. 2

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

[28] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 483–499, Cham, 2016. Springer International Publishing. 4

[29] Cristina Palmero, Javier Selva, Mohammad Ali Bagheri, and Sergio Escalera. Recurrent cnn for 3d gaze estimation using appearance and shape cues. *arXiv preprint arXiv:1805.03064*, 2018. 2

[30] Seonwook Park, Emre Aksan, Xucong Zhang, and Otmar Hilliges. Towards end-to-end video-based eye-tracking. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 747–763. Springer, 2020. 1, 2, 5, 6, 7

[31] Seonwook Park, Adrian Spurr, and Otmar Hilliges. Deep pictorial gaze estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 721–738, 2018. 1

[32] Narendra Patwardhan, Stefano Marrone, and Carlo Sansone. Transformers in the real world: A survey on nlp applications. *Information*, 14(4):242, 2023. 4

[33] Thammathip Piumsomboon, Gun Lee, Robert W Lindeman, and Mark Billinghurst. Exploring natural eye-gaze-based interaction for immersive virtual reality. In *2017 IEEE symposium on 3D user interfaces (3DUI)*, pages 36–39. IEEE, 2017. 1

[34] Rafael F Ribeiro and Paula DP Costa. Driver gaze zone dataset with depth data. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pages 1–5. IEEE, 2019. 1

[35] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986. 4

[36] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 4

[37] Kang Wang, Hui Su, and Qiang Ji. Neuro-inspired eye tracking with eye movement dynamics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9831–9840, 2019. 1

[38] Chenyang Zhang, Tiansu Chen, Eric Shaffer, and Elahe Soltanaghai. Focusflow: 3d gaze-depth interaction in virtual reality leveraging active visual depth manipulation. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery. 1

[39] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4511–4520, 2015. 1, 2

[40] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. It's written all over your face: Full-face appearance-based gaze estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 51–60, 2017. 1, 2, 6, 7

[41] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):162–175, 2017. 2, 6, 7

[42] Xiaolong Zhou, Jianing Lin, Jiaqi Jiang, and Shengyong Chen. Learning a 3d gaze estimator with improved itracker combined with bidirectional lstm. In *2019 IEEE international conference on Multimedia and expo (ICME)*, pages 850–855. IEEE, 2019. 1, 2

[43] Zhiwei Zhu and Qiang Ji. Novel eye gaze tracking techniques under natural head movement. *IEEE Transactions on biomedical engineering*, 54(12):2246–2260, 2007. 2