DVGaze: Dual-View Gaze Estimation

Yihua Cheng
Beihang University
Beijing, China
yihua_c@buaa.edu.cn

Feng Lu*
Beihang University
Beijing, China
lufeng@buaa.edu.cn

Abstract

Gaze estimation methods estimate gaze from facial appearance with a single camera. However, due to the limited view of a single camera, the captured facial appearance cannot provide complete facial information and thus complicate the gaze estimation problem. Recently, camera devices are rapidly updated. Dual cameras are affordable for users and have been integrated in many devices. This development suggests that we can further improve gaze estimation performance with dual-view gaze estimation. In this paper, we propose a dual-view gaze estimation network (DV-Gaze). DV-Gaze estimates dual-view gaze directions from a pair of images. We first propose a dual-view interactive convolution (DIC) block in DV-Gaze. DIC blocks exchange dual-view information during convolution in multiple feature scales. It fuses dual-view features along epipolar lines and compensates for the original feature with the fused feature. We further propose a dual-view transformer to estimate gaze from dual-view features. Camera poses are encoded to indicate the position information in the transformer. We also consider the geometric relation between dual-view gaze directions and propose a dual-view gaze consistency loss for DV-Gaze. DV-Gaze achieves state-of-the-art performance on ETH-XGaze and EVE datasets. Our experiments also prove the potential of dual-view gaze estimation. We release codes in https://github.com/yihuacheng/DVGaze.

1. Introduction

Human gaze provides important cues for understanding human cognition [25] and behavior [9]. It has applications in various fields such as salience detection [32, 8, 31], virtual reality [23, 20, 18] and human-computer interaction [11, 7].

Gaze estimation methods estimate human gaze from facial appearance. Conventional gaze estimation methods usually learn person-specific eye models and fit the eye model
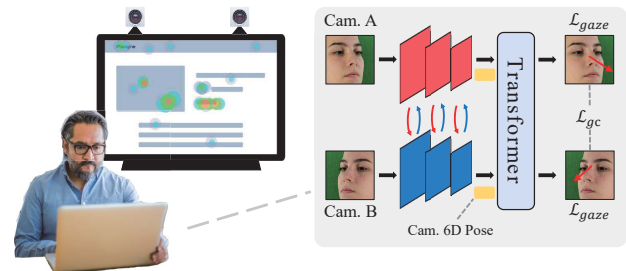
Figure 1. We explore dual-view gaze estimation in this work. We propose DV-Gaze to estimate gaze directions from a pair of images. DV-Gaze contains dual-view interactive convolution blocks which exchange the information of a pair of images during convolution in multiple feature scales, and a transformer to estimate gaze from dual-view feature.

to estimate human gaze. These model-based methods need to build specific camera system which contains multiple IR cameras and light sources [13]. Although model-based methods have good accuracy, the complex camera system brings high costs and harms flexibility. Appearance-based gaze estimation methods have low requirements in devices. It only requires a single webcam to capture facial appearance and directly learns a mapping function from the appearance to gaze. The low requirement means appearance-based gaze estimation methods have larger potential than model-based methods. They attract much attention and become a hotspot. However, the low requirement also brings limitations. In particular, one single webcam has a limited field of view and therefore captures incomplete facial appearance due to facial self-occlusion. The problem complicates gaze estimation and brings performance drop. To handle the problem, recent methods usually design efficient feature extraction networks[6, 29] or synthesize more images to cover data space[24, 30].

In this paper, we explore a new direction for gaze estimation. Recently, camera devices are rapidly developed and the cost of camera is also decreased. Dual cameras are affordable for users [12] and have been applied in many

devices [26]. The developing dual-camera devices make it possible and meaningful to estimate the human gaze with dual cameras. Compared with a single camera, dual cameras provide a larger field of view. Dual-view images can also provide more cues for gaze estimation. These advantages indicate dual-view gaze estimation can further improve gaze estimation accuracy than conventional single-view methods.

We propose a dual-view gaze estimation network (DV-Gaze) in this work. Common solutions usually extracts dual-view features from dual-view images and concatenate the two features for gaze estimation. They only fuse dual-view information in a high level. Our idea is to exchange dual-view information anywhere. We first propose a dual-view interactive convolution (DIC) block. DIC blocks exchange dual-view information during convolution. The block first fuses dual-view features along epipolar lines and add the fused features back to original features for compensation. It then performs feature extraction separately on the two compensated feature maps with convolution layers. We stack multiple DIC blocks to exchange dual-view information in multiple feature scales during convolution. We further propose a dual-view transformer. The transformer estimates dual-view gaze directions from dual-view features. We use camera pose to indicate the position information in the transformer. We also consider the geometric relation between dual-view gaze directions and propose a dual-view gaze consistency loss function.

Overall, we summarize our contributions as following.

1. We explore dual-view gaze estimation in this work. To the best of our knowledge, our work is the first to explore dual-view gaze direction estimation.

2. We propose DIC blocks for dual-view feature extraction. The block fuses dual-view features along epipolar lines and use fused features to compensate original features. Our method exchanges dual-view feature in multiple feature scales during convolution with DIC blocks.

3. We propose a dual-view transformer to estimate gaze from dual-view feature. We use camera pose to indicate the position information in the transformer. We also propose a dual-view gaze consistency loss which improves performance in a self-supervised manner.

2. Related work

2.1. Appearance-based Gaze Estimation

Appearance-based gaze estimation methods aim to estimate human gaze from facial appearance [5]. These methods usually learn a mapping function f from face or eye images $\mathbf{I}$ to human gaze $\mathbf{g}$, *i.e.*, $\mathbf{g} = f(\mathbf{I})$. Recently, deep learning shows good performance in many computer vision tasks. Gaze estimation with deep learning also attract much attention. Cheng *et al*. propose an asymmetric regression for gaze estimation from two eye images [4]. They build an asym-

metric network which adaptively assigns different weights for two-eye images. They also further fuse facial feature to improve performance [6]. Park *et al*. propose a pictorial representation for eye images [22]. They eliminate the identity difference between subjects by generating the pictorial representation and estimate gaze from the representation. Chen *et al*. use dilated convolutional network to capture the subtle changes in eye images [1]. Cheng *et al*. propose a coarse-to-fine network to integrate face and eye images [2]. They estimate basic gaze directions from face images and refine basic gaze directions with eye images. These methods all achieve great performance in many benchmarks. However, the performance of these methods still cannot meet some high accuracy requirements. An accurate gaze estimation model is always crucial and demanded.

2.2. Gaze Estimation with Multiple Cameras

Gaze estimation with multiple cameras is always a hot topic in conventional gaze estimation methods. These methods usually build multi-camera systems and estimate human gaze with geometric eye models [13]. Tonsen *et al*. embed multiple millimeter-sized RGB cameras into a normal glasses frame. They use multi-layer perceptrons to process the eye images captured by different cameras and concatenate the extracted feature to estimate gaze[27]. Common multi-camera gaze estimation systems are usually set in XR devices, such as Meta Quest Pro.

Deep learning based gaze estimation methods attract much attention in the last decade. However, most of methods are proposed for single-view gaze estimation. There are merely a few methods for multi-view gaze estimation. Lian *et al*. estimate points of gaze with three cameras [17] . They extract feature maps of three views and use a max-pooling layer between multi-views to extract feature. Kim *et al*. estimate gaze zones with three cameras [15]. They simply concatenate extracted feature of three views for prediction.

3. Preliminaries

3.1. Dual-View Gaze Estimation

We explore dual-view gaze estimation in this paper. We first define some notations and describe the task of dual-view gaze estimation.

We assume there are two cameras A and B. In the rest of this section, we use superscripts ^{A} and ^{B} to identify different cameras. The two cameras capture face images $\mathbf{I}^A$ and $\mathbf{I}^B$. Dual-view gaze estimation aims to learn a mapping function f from $\mathbf{I}^A$ and $\mathbf{I}^B$ to gaze directions $\mathbf{g}^A$ and $\mathbf{g}^B$, *i.e.*,

$$(\mathbf{g}^A, \mathbf{g}^B) = f(\mathbf{I}^A, \mathbf{I}^B). \tag{1}$$

The two gaze directions are separately defined in the camera coordinate systems (CCS) of dual cameras. It is obvious $\mathbf{g}^A$ and $\mathbf{g}^B$ can be converted to each other with camera rotation.
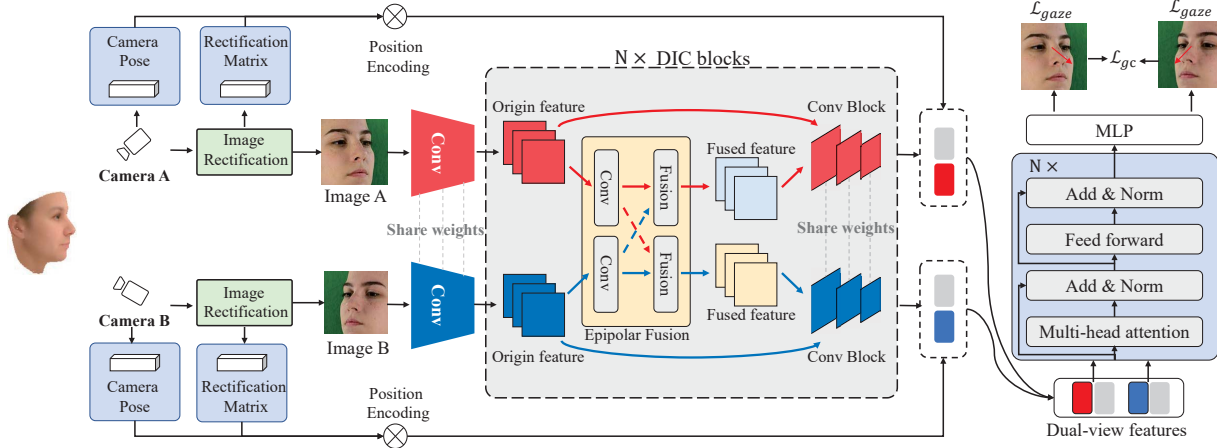
Figure 2. The pipeline of our dual-view gaze estimation. Two cameras capture dual-view images of human facial images. We first perform image rectification method on both images. We save the rectification matrix and origin camera pose for the rest steps. We input two rectified images into DV-Gaze. DV-Gaze first performs a primary convolution to extract dual-view features. We then send dual-view features into DIC blocks. The DIC block fuses dual-view feature along epipolar lines and add fused feature back to original features to mix information. We further perform feature extraction on the feature with convolution layers. We stack multiple DIC blocks to exchange dual-view information in different feature scales. We propose a dual-view transformer to estimate gaze from dual-view feature. We encode camera pose as the position feature in transformer. The loss function of DV-Gaze consists of common gaze loss and a self-supervised dual-view gaze consistency loss.

We also make following assumptions. 1) The 6D pose of two cameras are fixed and known. The pose can be obtained by camera calibration. We use $\mathbf{R}_c \in \mathbb{R}^{3 \times 3}$ to represent an rotation matrix and $\mathbf{t}_c \in \mathbb{R}^{3 \times 1}$ to represent a translation vector, *i.e.*, given a point p in the world coordinate system, we can convert it into CCS of camera A as $\mathbf{R}_c^A p + \mathbf{t}_c^A$. 2) We have head pose $\mathbf{R}_h \in \mathbb{R}^{3 \times 3}$ and $\mathbf{t}_h \in \mathbb{R}^{3 \times 1}$ in each view. The head pose can be obtained by existing head trackers [16].

3.2. Dual-View Images Rectification

A key information in dual-view system is the correspondence between two cameras. Epipolar lines indicate the correspondence that two correctly matched points should lie on their corresponding epipolar lines. In this work, we utilize the correspondence to fuse dual-view feature. To simplify the process of finding epipolar lines, we use image rectification methods [34] to make epipolar lines horizontal.

We briefly introduce the image rectification method in this section. We set a reference point in face images, *e.g.*, face centers. The image rectification method computes a transformation matrix $\mathbf{M} = \mathbf{SR}_{rec}$, where $\mathbf{S}$ is a scale matrix and $\mathbf{R}_{rec}$ is an rotation matrix. The scale matrix scales face images so that the distance between the reference point and virtual cameras is constant. The rotation matrix transforms face images via perspective transformation. It ensures virtual cameras point at the reference point, *i.e.*, face centers are located in image centers, and cancel the row axis of head pose. Please refer [5] for more technical details.

After the image rectification, epipolar lines are horizontal

and lie on corresponding rows. Meanwhile, the methods perform perspective transformation for the rectification. This operation changes virtual camera pose of images. The new camera pose can be computed by $\mathbf{R}_{rec}\mathbf{R}_c$.

4. Dual-View Gaze Estimation

4.1. Overview

In this section, we propose a dual-view gaze estimation network (DV-Gaze). We show the pipeline of DV-Gaze in Fig. 2. We first use a convolution layer to extract dual-view features from dual-view images. We then propose DIC blocks. The block fuses dual-view feature maps along epipolar lines and uses fused feature to compensate origin feature. It performs feature extraction on the compensated feature with convolution layers. We stack multiple DIC blocks to exchange dual-view information in multiple feature scales. We also propose a dual-view transformer to estimate gaze from dual-view feature. We use camera pose to indicate the position information in the transformer. DV-Gaze predicts dual-view gaze directions from dual-view images. We use common gaze loss, *i.e.* L1 loss, as loss function. We also propose a self-supervised dual-view gaze consistency loss based on the geometric relation of dual-view gaze.

In the rest of this section, we first introduce the DIC block and the dual-view transformer. We then describe DV-Gaze in detail. We finally show all loss functions of DV-Gaze.
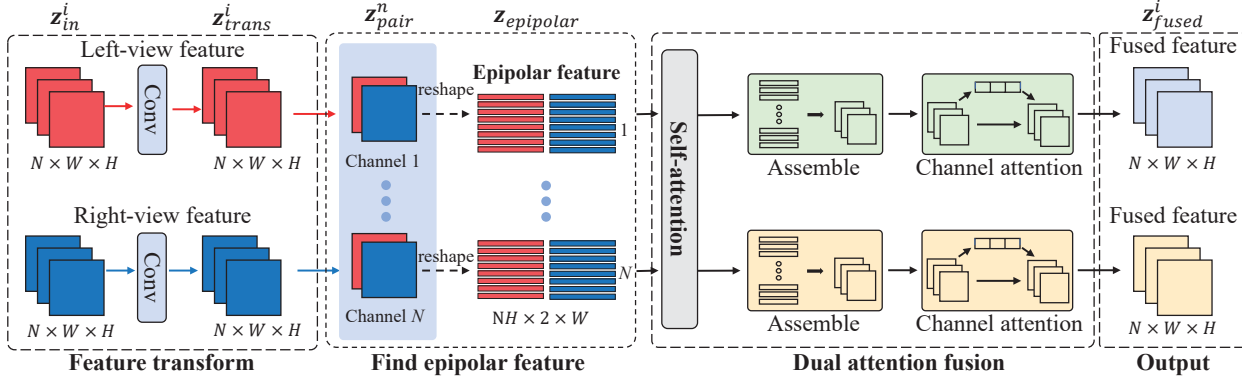
Figure 3. The pipeline of epipolar fusion. We first perform a feature transform on origin feature. To reduce computational resource, we use the bottleneck architecture of residual block for the feature transform. We then aggregate feature maps and find epipolar feature in channels. Due to the image rectification in data pre-processing, the epipolar feature is the corresponding rows in two feature maps. We use a self-attention mechanism to fuse epipolar feature and assemble feature vectors into feature maps. We also apply a channel attention in the feature maps. The output contains two feature maps which have the same size as input feature.

4.2. Dual-View Interactive Convolution Block

The two cameras in the dual-view system have different camera positions. This means the image of each view would miss some information about human gaze. It is intuitive that the performance would be improved if we exchange dual-view information to compensate the missing information.

We propose DIC blocks to exchange dual-view information during convolution. It fuses dual-view feature and adds the fused features back to the original feature to mix information. Simply speaking, common convolution blocks extracts feature as

$$\mathbf{z}_{\text{out}}^i = \text{Conv}(\mathbf{z}_{\text{in}}^i), i \in \{A, B\} \qquad (2)$$

where $\mathbf{z}_{\text{in}}^i \in \mathbb{R}^{N \times H \times W}$ is the origin feature of each view and $\mathbf{z}_{\text{out}}^i$ is the extracted feature.

DIC blocks aim to fuse dual-view features $\{\mathbf{z}_{\text{in}}^i\}$ to learn dual-view fused features $\{\mathbf{z}_{\text{fused}}^i \in \mathbb{R}^{N \times H \times W}\}$ to compensate $\mathbf{z}_{\text{in}}^i$. The formulation of DIC block is

$$\mathbf{z}_{\text{out}}^i = \text{Conv}\left(\mathbf{z}_{\text{in}}^i + \mathbf{z}_{\text{fused}}^i\right), \qquad (3)$$

DIC blocks contain three steps to learn fused features. We show the pipeline in Fig. 3.

1) Feature transform. We first use two convolution blocks to respectively transform dual-view features. We send $\{\mathbf{z}_{\text{in}}^i\}$ into residual blocks [14] and get feature maps $\{\mathbf{z}_{\text{trans}}^i\}$, where $\mathbf{z}_{\text{trans}}^i$ has the same dimension as $\mathbf{z}_{in}^i$. To reduce computational resource, we add a bottleneck in the residual block. Overall, the residual block contains three convolution layers, where the kernel size is set as 3×3 for the second layer and 1×1 for the first and third layers. The number of kernel is respectively set as $\left(\frac{N}{4}, \frac{N}{4}, N\right)$.

2) Find epipolar feature. Epipolar lines indicate the correspondence between dual cameras. DIC blocks aim to fuse dual-view features along epipolar lines. We define a pair of features along epipolar lines as epipolar features. In this step, our aim is to find epipolar feature from $\{\mathbf{z}_{\text{trans}}^i\}$.

We aggregate feature maps $\mathbf{z}_{\text{trans}}^A$ and $\mathbf{z}_{\text{trans}}^B$ in channels. As a result, we have N pairs of feature maps. We denote a pair of feature maps as $\mathbf{z}_{\text{pair}}^n \in \mathbb{R}^{2 \times H \times W}$, where $n \in \{1...N\}$. The next goal is to find epipolar feature from $\mathbf{z}_{\text{pair}}^n$.

As described in Sec. 3.2, we rectify images in data pre-processing. It means epipolar lines are horizontal and lie on corresponding rows. Therefore, we sample epipolar features from the corresponding rows of $\mathbf{z}_{\text{pair}}^n$, *i.e.* $\mathbf{z}_{\text{pair}}^n$ contains H epipolar features and the shape of each epipolar feature is $2 \times W$. We sample a total of NH epipolar features from $\{\mathbf{z}_{\text{pair}}^n\}$. We define all epipolar features as $\mathbf{z}_{\text{epipolar}} \in \mathbb{R}^{NH \times 2 \times W}$.

3) Dual attention fusion. Our goal is to learn a mapping function $\psi : \mathbb{R}^{2 \times W} \to \mathbb{R}^W$. The mapping function learns fused features from epipolar features.

We use a self-attention module [28] to fuse epipolar features. We use two MLP layers to learn the query and the key from epipolar features and directly set the value as epipolar features. The self-attention module adaptively learns weights from the query and the key. The fused feature is the weighted-sum of the value. We reassemble all fused feature vectors and get fused feature maps $\mathbf{z}_{\text{fused}} \in \mathbb{R}^{N \times H \times W}$. Note that, the self-attention module outputs two fused feature, *i.e.*, the self-attention is a mapping function, $\mathbb{R}^{2 \times W} \to \mathbb{R}^{2 \times W}$. We respectively reassemble them and get $\mathbf{z}_{\text{fused}}^A$ and $\mathbf{z}_{\text{fused}}^B$.

We also add a channel attention module after the self-attention module. We input $\mathbf{z}_{\text{fused}}^A$ and $\mathbf{z}_{\text{fused}}^B$ into an average pooling layer and use the output feature to learn weights for each channel. We multiply the weights with the two fused features and get final fused features. We slightly abuse the notation here and also define the final fused feature as $\mathbf{z}_{\text{fused}}^A$ and $\mathbf{z}_{\text{fused}}^B$. We use the two feature to respectively

compensate original dual-view feature as Eq. (3).

4.3. Dual-View Transformer

We use a transformer encoder to estimate gaze from dual-view feature. We assume the dual-view features as $\left\{\mathbf{z}_{\text{vec}}^i\right\}$. We can formulate this module as

$$(\hat{\mathbf{g}}^A, \hat{\mathbf{g}}^B) = \text{Transformer}(\mathbf{z}_{\text{vec}}^A, \mathbf{z}_{\text{vec}}^B), \qquad (4)$$

where we use $\hat{\mathbf{g}}^i$ to represent the predicted value. Besides, The camera pose is an important information in the dual-view camera system. It indicates the relations between dual-view images. We also input camera pose into the transformer for gaze estimation. We consider transformers usually utilize position information to identify input features, *e.g.*, position embedding in ViT[10]. We use the camera pose to indicate the position information in our work.

More concretely, we use camera orientation and translation vectors to represent position feature. As described in the preliminaries section, we have camera positions $\left\{\mathbf{R}_c^i, \mathbf{t}_c^i\right\}$ via calibration and the rotation matrix $\left\{\mathbf{R}_{\text{rec}}^i\right\}$ in image rectification. Note that $\left\{\mathbf{R}_{\text{rec}}^i\right\}$ is different for each image. We multiply $\mathbf{R}_{\text{rec}}^i$ and $\mathbf{R}_c^i$ to get the rotation matrix of virtual cameras and use the third column in the rotation matrix to represent camera orientation.

$$\mathbf{z}_{\text{rot}}^i = \mathbf{R}_{\text{rec}}^i \mathbf{R}_c^i [:, 2] \qquad (5)$$

Let $\mathbf{z}_{pos}$ for the position feature, we have

$$\mathbf{z}_{\text{pos}}^i = [\mathbf{z}_{\text{rot}}^i; \mathbf{t}_c^i], \qquad (6)$$

where $[\quad]$ is a concatenation operation and $\mathbf{z}_{\text{pos}}^i \in \mathbb{R}^6$.

Besides, the position features in transformers usually are discrete values. Transformers use position embedding to expand the position feature to a large dimension [10]. However, $\mathbf{z}_{\text{pos}}^i$ is a continuous vector and it is not applicable to use the position embedding. In our work, we follow NeRF [19] and use multi-MLP layers to expand $\mathbf{z}_{pos}^i$. We first expand the dimension with positional encoding which enables the position feature to contain more high frequency information. The positional encoding is

$$\gamma(p) = \left\{\sin(2^k \pi p), \cos(2^k \pi p), p\right\}_{k=0}^L, \qquad (7)$$

where L is length of the encoding.

We respectively apply the function into each dimension of $\mathbf{z}_{\text{pos}}^i$. The positional encoding can also be represented as a function $\Phi : \mathbb{R}^6 \to \mathbb{R}^{6*(3L)}$. We then use a multi-MLP layers $\Pi : \mathbb{R}^{6*(3L)} \to \mathbb{R}^d$ to learn the final position feature, where d is the length of $\mathbf{z}_{\text{vec}}^i$. The whole process can be formulated as

$$\mathbf{z}_{\text{pos\_exp}}^i = \Pi(\Phi(\mathbf{z}_{\text{pos}}^i)) \qquad (8)$$

We input dual-view feature vectors $\left\{\mathbf{z}_{\text{vec}}^i\right\}$ with corresponding position feature $\left\{\mathbf{z}_{\text{pos\_exp}}^i\right\}$ into transformers and use a MLP head to predict gaze.
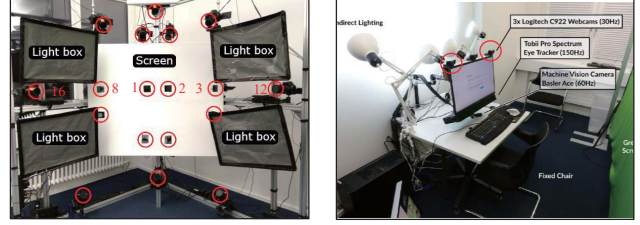

Figure 4. We show the camera position in ETH-XGaze (left)[33] and EVE (right) [21] dataset. We use the No.3 and No.8 cameras in ETH-XGaze and the two cameras on the edge of screen in EVE.

4.4. Dual-View Gaze Estimation Network

We propose dual-view gaze estimation network in this section. The architecture of DV-Gaze is shown in Fig. 2. Two cameras capture human facial appearance and provide two facial images. We first perform image rectification in the two images and input rectified dual-view images into DV-Gaze. Although we propose DIC block to fuse dual-view feature during convolution, it is unreasonable that we directly fuse raw dual-view images. Therefore, we first use a convolution layer to respectively extract primary dual-view feature maps from images. We then send the dual-view feature maps into DIC blocks for feature extraction. We stack multiple DIC blocks and obtain the output of the final DIC block. The outputs are dual-view feature maps. We use an average pooling layer and a MLP layer to map the two feature maps into the dual-view feature vectors $\mathbf{z}_{\text{vec}}^A$ and $\mathbf{z}_{\text{vec}}^B$. We finally feed the dual-view feature vectors into the dual-view transformer to estimate dual-view gaze directions.

The DV-Gaze estimates gaze $\hat{\mathbf{g}}^A$ and $\hat{\mathbf{g}}^B$ from dual-view images, where $\hat{\mathbf{g}}^i \in \mathbb{R}^2$ and contains pitch and yaw. We use L1 loss to measure the loss of gaze estimation

$$\mathcal{L}_{gaze} = \sum_{i \in \{A,B\}, j \in \{1...K\}} \left| \mathbf{g}_j^i - \hat{\mathbf{g}}_j^i \right|, \qquad (9)$$

where K denotes the number of all data.

Dual-view gaze consistency loss. In dual-view gaze estimation, the gaze directions of dual views can be converted into each other if camera positions are known. Therefore, we design a self-supervised dual-view gaze consistency loss. The loss requires dual-view gaze directions are the same when we convert them into the same coordinate system. In detail, we convert $\{\hat{\mathbf{g}}^i\}$ into the world coordinate system and use MSE loss to measure the error. The gaze consistency loss is computed as

$$\mathcal{L}_{gc} = \left| \left(\mathbf{R}_{\text{rec}}^A \mathbf{R}_c^A\right)^{-1} \hat{\mathbf{g}}_{\text{3d}}^A - \left(\mathbf{R}_{\text{rec}}^B \mathbf{R}_c^B\right)^{-1} \hat{\mathbf{g}}_{\text{3d}}^B \right|_2, \quad (10)$$

where $\hat{\mathbf{g}}_{\text{3d}}^i$ denotes gaze directions in euclidean coordinates. $\hat{\mathbf{g}}_{\text{3d}}^i = (x, y, z)$ and $\hat{\mathbf{g}}^i = (\phi, \theta)$ can be converted as

Table 1. We compare our method with single-view gaze estimation methods. We evaluate compared methods in each view and show results in the first row. We also expand training set where it contains dual-view images. We train models in the expand set and show results in the second row. We mark the highest result among compared methods for convenience.

	Methods	ETH-XGaze		EVE	
		Left	Right	Left	Right
Full set	Full Face [35]	5.11	4.77	4.23	5.06
	GazeTR [3]	3.46	3.48	3.18	3.72
	CA-Net [2]	-	-	3.03	3.81
	Dilated-Net [1]	-	-	3.10	3.46
Expand set	Full Face [35]	4.98	4.89	4.59	4.60
	GazeTR [3]	3.11	3.49	2.92	3.81
	CA-Net [2]	-	-	3.31	3.87
	Dilated-Net [1]	-	-	3.31	3.39
	DV-Gaze	**2.27**	**2.87**	**2.57**	**2.68**

Table 2. We compare our method with multi-view gaze estimation methods. The column of Avg. means we average dual-view gaze directions in the world coordinate system and report the angular error between averaged gaze and ground truth. We mark the highest performance among compared methods.

Methods	ETH-XGaze			EVE		
	Left	Right	Avg.	Left	Right	Avg.
MMGE [15]	6.44	7.47	6.88	5.71	8.19	6.19
MGT [17]	2.86	3.33	2.97	3.37	3.76	3.02
Res18	2.61	3.04	2.70	3.39	3.58	2.97
Trans	3.41	3.15	2.74	3.76	3.50	2.98
DV-Gaze	**2.27**	**2.87**	**2.46**	**2.57**	**2.68**	**2.49**

$$\begin{cases} x = -\cos(\phi) * \sin(\theta) \\ y = -\sin(\phi) \\ z = -\cos(\phi) * \sin(\theta) \end{cases} \quad (11)$$

The final loss function is the weighted-sum of gaze loss and gaze consistency loss.

$$\mathcal{L} = \alpha\mathcal{L}_{gaze} + \beta\mathcal{L}_{gc} \quad (12)$$

5. Experiment

5.1. Datasets

We conduct experiments on ETH-XGaze [33] and EVE datasets [21]. The two datasets both provide large-scale multi-view images. We show the camera layout of the two datasets in Fig. 4. Our method estimates human gaze from dual-view cameras. Therefore, we use the No.3 and No.8 cameras in ETH-XGaze dataset and the two cameras on the edge of screen in EVE dataset. We also denote them as left and right views in the rest experiment.

Data pre-processing. The two datasets both split origin data into training, test and validation sets. They align face images with the rectification methods[34] and provide the rectified data. Therefore, we directly use the provided images in ETH-XGaze dataset for experiments. EVE dataset provides rectified video clips. We sample one image per fifteen frames from video clips to construct evaluation set.

On the other hand, gaze directions are usually calculated by $\mathbf{g} = o_t - o_s$ in data collection, where o_s represents the 3D position of gaze origin and o_t represents the 3D position of gaze targets. In the two datasets, they define face centers as the origin of gaze. To calculate the 3D position of face

centers, they fit a 3D morphable face model and estimate 3D landmarks with the EPnP method [16]. This pipeline is reasonable for single-view gaze estimation while causes inconsistent gaze between different views. Therefore, we re-calculate gaze directions in the two datasets. More concretely, given a pair of images, we get the corresponding 2D facial landmarks from datasets and estimate 3D landmarks using triangulation. We select nose as gaze origin since nose is usually visible in all views. We compute new gaze directions with the new gaze origin.

Besides, the two datasets do not provide gaze targets annotation of test test. Therefore, we also re-define the dataset splits. We split the origin training set into two subsets and define the two subsets as training and validation sets. We use the original validation set as test set in our experiments.

Evaluation metric: We use angular error as the evaluation metric, where a smaller error represents a better model.

5.2. Comparison with Single-View Methods

Single-view gaze estimation is the main track in gaze estimation field. Recently, many advanced methods are proposed for single-view gaze estimation. We first conduct comparison with them to show the advantage of dual-view gaze estimation. We select FullFace [35], GazeTR [3], CA-Net [2] and Dilated-Net [1] for comparison. We evaluate these methods on ETH-XGaze and EVE datasets. Note that, CA-Net and Dilated-Net require eye images for evaluation while ETH-XGaze cannot always provides reliable eye images due to large head pose. We do not evaluate the two methods in ETH-XGaze dataset.

We first evaluate these methods in each view and show the performance in the first row of Tab. 1. The result shows most of methods have $\sim 3°$ accuracy in two datasets. DV-Gaze brings large performance improvement compared with these methods. Compared with the highest performance among compared methods, our method brings $10\% \sim 30\%$ improvement in the two datasets. This shows the advantage

Table 3. We conduct ablation study about the dual-view transformer and the DIC block. Note that, the ablation of DIC blocks means we do not use Eq. (3) but Eq. (2) in DV-Gaze.

Dual-view Trans.	DIC Block	ETH-XGaze			EVE		
		Left	Right	Avg.	Left	Right	Avg.
$\times$	$\times$	2.61	3.04	2.70	3.39	3.58	2.97
$\checkmark$	$\times$	2.40	2.99	2.51	2.97	3.11	2.83
$\checkmark$	$\checkmark$	**2.27**	**2.87**	**2.46**	**2.57**	**2.68**	**2.49**

Table 4. We perform ablation study to show the effectiveness of each module. w/o $\mathbf{z}_{\mathrm{pos\_exp}}$ means we do not add camera positions into dual-view transformers. w/o $\mathbf{z}_{\mathrm{in}}$ means we do not use fused feature to compensate origin dual-view feature in Eq. (3). w/o $\mathcal{L}_{gc}$ means we remove the dual-view gaze consistency loss.

	ETH-XGaze			EVE		
	Left	Right	Avg.	Left	Right	Avg.
w/o $\mathbf{z}_{\mathrm{pos\_exp}}$	2.49	3.54	2.70	3.00	3.21	2.71
w/o $\mathbf{z}_{\mathrm{in}}$	2.62	2.84	2.69	2.67	2.93	2.71
w/o $\mathcal{L}_{gc}$	2.58	3.08	2.56	2.78	2.78	2.55
DV-Gaze	**2.27**	**2.87**	**2.46**	**2.57**	**2.68**	**2.49**

of dual-view gaze estimation.

We also expand the training set of compared methods. DV-Gaze uses dual-view images for training while compared methods only use single-view images for training. The difference maybe bring unfair comparison. Therefore, we expand the training set of compared method where it contains dual-view images. We train compared methods in the expanded set and report the performance of each view in the second row of Tab. 1. Most of methods has performance improvement with the expanded set. This is because the expanded set contains more images. However, we notice that these improvements are all small. The reason is they do not utilize dual-view relations. Our method also has the best performance in all datasets.

5.3. Comparison with Multi-View Methods

We conduct comparison with multi-view methods in this section. MGT [17] estimates points of gaze from multi-view images. The method uses eye images as input. We change the input to face images for fair comparison. MMGE [15] estimates gaze zone from multi-view images. The model is simple and only contains three convolution blocks. We modify the last MLP layer to estimate gaze directions. We show their performance in Tab. 2. Besides, dual-view gaze directions can be converted to each other via camera rotation. We rotate dual-view gaze directions into the world coordinate system and average them to get one unique gaze direction. We count the performance of the gaze direction and show results in the column of Avg. The result shows DV-Gaze has better performance than both MGT and MMGE.

We also build dual-view gaze estimation networks with strong performing architecture for extensive comparison. We use ResNet-18 [14] to extract feature from dual-view images and concatenate dual-view features to estimate dual-view gaze. We denote the method as *Res18*. We also use ResNet-18 to extract feature and use a 6-layer transformer to fuse dual-view features[3] We estimate dual-view gaze from the output of transformer. The method is denoted as *Trans*. We show the performance of the two methods in Tab. 2. The two methods both have good performance since they uses strong performing architecture. However, our method also has the best performance compared with them. The result proves

our method is effective in dual-view gaze estimation.

5.4. Ablation Study

We propose DIC blocks and dual-view transformer for dual-view gaze estimation. We conduct ablation study to demonstrate the advantage of them. We first evaluate the basic model of DV-Gaze. The basic model uses convolution layers of ResNet18 to extract feature from dual-view images. It concatenates dual-view features and estimates gaze with a MLP layer. We then add the dual-view transformer into the basic model. The model uses dual-view transformer to estimate gaze from dual-view feature. We also further add DIC blocks into the model, *i.e.*, DV-Gaze. The model uses DIC blocks to exchange dual-view information during convolution. The performance is shown in Tab. 3. The result shows both the DIC block and the dual-view transformer bring sufficient performance improvement.

Besides, DV-Gaze also contains some special mechanisms. We further evaluate these mechanisms in detail. 1) DV-Gaze adds camera position into the dual-view transformer. We ablate the camera position and denote the experiment as w/o $\mathbf{z}_{\mathrm{pos\_exp}}$. 2) In DIC blocks, we do not use fused feature as output but use the fused feature to compensate origin feature, *i.e.*, Eq. (3). We also ablate $\mathbf{z}_{\mathrm{in}}$ from Eq. (3). 3) We ablate the dual-view gaze consistency loss from loss function. We show results of the three experiments on Tab. 4. The result proves the advantage of the three mechanisms.

5.5. Different Camera Pairs

In the previous experiment, we evaluate camera pairs in ETH-XGaze and EVE dataset, where two cameras are respectively placed on the one side of a screen. We also evaluate more camera pairs in ETH-XGaze for deep insights. We conduct two experiments, *Small Distance* and *Long Distance*. In the *Small Distance*, we select No.1 camera and No.2 camera as the camera pairs. In the *Long Distance*, we select No.12 camera and No.16 camera as the camera pairs. We also define the original camera pairs in ETH-XGaze as *Medium Distance*. We evaluate the performance of GazeTR
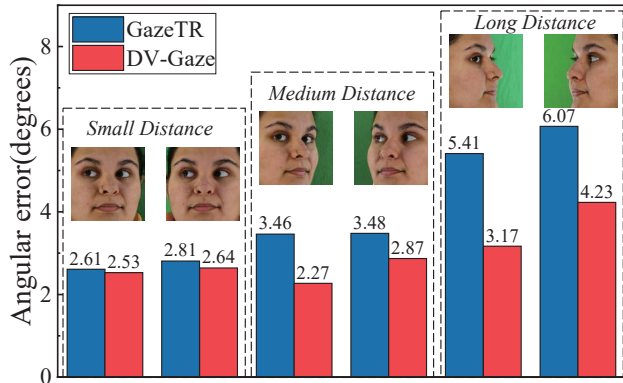
Figure 5. We select different camera pairs in ETH-XGaze and conduct more evaluations for DV-Gaze. We also evaluate GazeTR and show the performance for comparison. It is obvious that larger camera distance brings more pixel difference between a pair of images. One eye region is not visible in the *Long Distance* and DV-Gaze has the highest performance improvement in such case. On the other hand, it is interesting that camera pairs in *Small distance* capture similar facial images while DV-Gaze can also bring performance improvement to some extent.

for reference and show all results in Fig. 5. We also paste captured images of the corresponding view below the figure.

It is obvious that larger camera distance brings more vision differences in dual-view images. Meanwhile, DV-Gaze can also bring more performance improvement. In the *Small Distance*, the captured images have no significant difference due to the small distance between camera pairs. It means the dual-camera system cannot provide extra vision information such as occlusion region than single-camera systems. However, the result shows DV-Gaze also has performance improvement in the such case despite the improvement is relatively small. In the *Medium Distance*, the eye region has a small occlusion in each images. DV-Gaze can bring $\sim 1°$ performance improvement than GazeTR. In the *Long Distance*, it is obvious that one eye region is not visible in images. DV-Gaze brings large improvement in the such case.

5.6. Select from Image Pairs

We propose dual-view gaze estimation in this work. Different from conventional single-view methods, DV-Gaze fuses dual-view feature to estimate human gaze. In this experiment, we define an oracle baseline where we select the best view from a pair of images and count the average result.

We first preform the selection on GazeTR. As shown in Tab. 5, we have $2.38°$ in ETH-XGaze and $2.50°$ in EVE. In fact, this result shows the upper bound of GazeTR which means we need to develop a very strong selection algorithm. Compared with GazeTR, DV-Gaze can easily reach the upper bound with dual-view fusion feature. DV-Gaze has $2.27°$ in the right view of ETH-XGaze and $2.57°$ in the left view of EVE. This result shows the advantage of dual-view gaze

Table 5. We obtain the results of GazeTR which is trained on each camera. We manually select the best view from each dual-view image pairs and count the average result. The result can be thought as the upper bound of the single-view method. DV-Gaze not only achieves better result than the upper bound in ETH-XGaze but also improves the upper bound in both two datasets.

	ETH-XGaze			EVE		
	Left	Right	Select	Left	Right	Select
GazeTR	3.49	3.11	2.38	2.92	3.81	2.50
DV-Gaze	2.87	2.27	2.00	2.57	2.68	2.21

estimation. Besides, we also perform the selection based on the result of DV-Gaze. It demonstrates DV-Gaze can also improve the upper bound with dual-view feature fusion.

6. Discussion

In this work, we have investigated the potential of dual-view gaze estimation. We conduct experiments and prove 1) Dual-view feature fusion is necessary and useful. 2) Dual-view gaze estimation methods could achieve better performance than single-view gaze estimation methods. These conclusions demonstrate the advantage of dual-view gaze estimation, and prove it has more potential compared with single-view gaze estimation. We also show the performance of DV-Gaze in different camera pairs. DV-Gaze brings performance improvement in all camera pairs. It indicates dual-view gaze estimation can be useful in many scenarios, such as laptop, desktop computer, smart screen, intelligent vehicles and etc. On the other hand, dual-view gaze estimation also brings new challenges, *e.g*., How to select the best view? Tab. 5 shows we can get a better performance if we can correctly select the best view from a pair of images. Dual-view images also provide extra stereo information. Can we utilize the information for further performance improvement?

7. Conclusion

We consider the limitation of single-view gaze estimation and explore dual-view gaze estimation in this work. We propose a dual-view gaze estimation network, DV-Gaze. DV-Gaze consists of convolutional layers, DIC blocks and a dual-view transformer. DIC blocks fuse dual-view features along epipolar lines and compensate for original features. We stack DIC blocks to fuse dual-view feature in multiple levels during convolution. We feed the output of the final DIC block into the dual-view transformer. The transformer encodes dual-view features and camera positions to estimate dual-view gaze directions. We conduct experiments on ETH-XGaze and EVE dataset. The experiment shows DV-Gaze has the best performance among compared methods. Our method reveals a potential direction in gaze estimation.

References

[1] Zhaokang Chen and Bertram E. Shi. Appearance-based gaze estimation using dilated-convolutions. In C.V. Jawahar, Hongdong Li, Greg Mori, and Konrad Schindler, editors, *The Asian Conference on Computer Vision*, pages 309–324, 2019. 2, 6

[2] Yihua Cheng, Shiyao Huang, Fei Wang, Chen Qian, and Feng Lu. A coarse-to-fine adaptive network for appearance-based gaze estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 2, 6

[3] Yihua Cheng and Feng Lu. Gaze estimation using transformer. *ICPR*, 2022. 6, 7

[4] Yihua Cheng, Feng Lu, and Xucong Zhang. Appearance-based gaze estimation via evaluation-guided asymmetric regression. In *The European Conference on Computer Vision*, 2018. 2

[5] Yihua Cheng, Haofei Wang, Yiwei Bao, and Feng Lu. Appearance-based gaze estimation with deep learning: A review and benchmark. *arXiv preprint arXiv:2104.12668*, 2021. 2, 3

[6] 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, 2

[7] Myungguen Choi, Daisuke Sakamoto, and Tetsuo Ono. Kuiper belt: Utilizing the "out-of-natural angle" region in the eye-gaze interaction for virtual reality. In *CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2022. 1

[8] Eunji Chong, Nataniel Ruiz, Yongxin Wang, Yun Zhang, Agata Rozga, and James M. Rehg. Connecting gaze, scene, and attention: Generalized attention estimation via joint modeling of gaze and scene saliency. In *The European Conference on Computer Vision*, 2018. 1

[9] Philipe Ambrozio Dias, Damiano Malafronte, Henry Medeiros, and Francesca Odone. Gaze estimation for assisted living environments. In *The IEEE Winter Conference on Applications of Computer Vision*, 2020. 1

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 5

[11] Carlos Elmadjian and Carlos H Morimoto. Gazebar: Exploiting the midas touch in gaze interaction. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7, 2021. 1

[12] Ana Belén González and Jose Pozo. The industrial camera modules market: Market review and forecast until 2022. *PhotonicsViews*, 16(2):24–26, 2019. 1

[13] E. D. Guestrin and M. Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on Biomedical Engineering*, 53(6):1124–1133, 2006. 1, 2

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 4, 7

[15] Jung-Hwa Kim and Jin-Woo Jeong. A preliminary study on performance evaluation of multi-view multi-modal gaze estimation under challenging conditions. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–7, 2020. 2, 6, 7

[16] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155–166, 2009. 3, 6

[17] D. Lian, L. Hu, W. Luo, Y. Xu, L. Duan, J. Yu, and S. Gao. Multiview multitask gaze estimation with deep convolutional neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 30(10):3010–3023, Oct 2019. 2, 6, 7

[18] Katerina Mania, Ann McNamara, and Andreas Polychronakis. Gaze-aware displays and interaction. In *ACM SIGGRAPH 2021 Courses*, pages 1–67, 2021. 1

[19] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 5

[20] Yun Suen Pai, Benjamin Tag, Benjamin Outram, Noriyasu Vontin, Kazunori Sugiura, and Kai Kunze. Gazesim: simulating foveated rendering using depth in eye gaze for vr. In *ACM SIGGRAPH 2016 Posters*, pages 1–2, 2016. 1

[21] Seonwook Park, Emre Aksan, Xucong Zhang, and Otmar Hilliges. Towards end-to-end video-based eye-tracking. In *The European Conference on Computer Vision*, pages 747–763. Springer, 2020. 5, 6

[22] Seonwook Park, Adrian Spurr, and Otmar Hilliges. Deep pictorial gaze estimation. In *The European Conference on Computer Vision*, 2018. 2

[23] Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions on Graphics (TOG)*, 35(6):1–12, 2016. 1

[24] Jiawei Qin, Takuru Shimoyama, and Yusuke Sugano. Learning-by-novel-view-synthesis for full-face appearance-based 3d gaze estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4981–4991, June 2022. 1

[25] Rima-Maria Rahal and Susann Fiedler. Understanding cognitive and affective mechanisms in social psychology through eye-tracking. *Journal of Experimental Social Psychology*, 85:103842, 2019. 1

[26] Simranpal Singh. Best smartphones with dual selfie camera. https://www.gizmochina.com/2020/01/18/best-smartphones-dual-selfie-camera/. 2

[27] Marc Tonsen, Julian Steil, Yusuke Sugano, and Andreas Bulling. Invisibleeye: Mobile eye tracking using multiple low-resolution cameras and learning-based gaze estimation. *Proceedings of the ACM on Interactive Mobile Wearable and Ubiquitous Technologies*, 1(3), Sept. 2017. 2

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 4

[29] Hengfei Wang, Jun O. Oh, Hyung Jin Chang, Jin Hee Na, Minwoo Tae, Zhongqun Zhang, and Sang-Il Choi. Gazecaps: Gaze estimation with self-attention-routed capsules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2668–2676, June 2023. 1

[30] Hengfei Wang, Zhongqun Zhang, Yihua Cheng, and Hyung Jin Chang. High-fidelity eye animatable neural radiance fields for human face. *arXiv preprint arXiv:2308.00773*, 2023. 1

[31] Wenguan Wang and Jianbing Shen. Deep visual attention prediction. *IEEE Transactions on Image Processing*, 27(5):2368–2378, 2018. 1

[32] Wenguan Wang, Jianbing Shen, Xingping Dong, Ali Borji, and Ruigang Yang. Inferring salient objects from human fixations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 1

[33] Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In *The European Conference on Computer Vision*, 2020. 5, 6

[34] Xucong Zhang, Yusuke Sugano, and Andreas Bulling. Revisiting data normalization for appearance-based gaze estimation. In *Proceedings of the ACM Symposium on Eye Tracking Research & Applications*, 2018. 3, 6

[35] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. It's written all over your face: Full-face appearance-based gaze estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2299–2308, 2017. 6