Unsupervised Gaze Representation Learning from Multi-view Face Images

Yiwei Bao  Feng Lu*
State Key Laboratory of VR Technology and Systems, School of CSE, Beihang University
{baoyiwei, lufeng}@buaa.edu.cn

Abstract

Annotating gaze is an expensive and time-consuming endeavor, requiring costly eye-trackers or complex geometric calibration procedures. Although some eye-based unsupervised gaze representation learning methods have been proposed, the quality of gaze representation extracted by these methods degrades severely when the head pose is large. In this paper, we present the Multi-View Dual-Encoder (MV-DE), a framework designed to learn gaze representations from unlabeled multi-view face images. Through the proposed Dual-Encoder architecture and the multi-view gaze representation swapping strategy, the MV-DE successfully disentangles gaze from general facial information and derives gaze representations closely tied to the subject’s eye-ball rotation without gaze label. Experimental results illustrate that the gaze representations learned by the MV-DE can be used in downstream tasks, including gaze estimation and redirection. Gaze estimation results indicate that the proposed MV-DE displays notably higher robustness to uncontrolled head movements when compared to state-of-the-art (SOTA) unsupervised learning methods.

1. Introduction

Vision is one of the most important sense for humans. Human gaze reveals the direction of visual attention, which is an important cue for understanding how humans perceive the surrounding world. Thus, gaze estimation techniques have become an vital tool in numerous applications, such as Virtual Reality and Augmented Reality[2, 30, 34], automotive safety [13, 26, 27] and healthcare [3, 20, 23]. In recent years, appearance-based gaze estimation methods have drawn a lot of attention, since these methods only require simple web cameras, eliminating the need for expensive eye trackers with dedicated devices such as infrared cameras. Among these methods, Convolutional Neural Networks (CNN) based approaches exhibit exceptional performance in unconstrained environments.

*Corresponding Author. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 62372019.

CNN-based gaze estimation methods are usually trained in an end-to-end manner with substantial amount of labeled data. The performance of CNN-based gaze estimation methods highly relies on the quantity and diversity of the training dataset. Unfortunately, annotating gaze directions is difficult. Unlike common Computer Vision tasks like Object Detection, gaze cannot be reliably annotated without specialized hardware. Gaze direction is either measured by costly eye trackers under controlled environment [29], or obtained by the line connecting the 3D face center to the 3D position of the gaze target [39]. It takes great effort and time to collect diverse labeled training data, since the participants need to stare at large amount of gaze targets as instructed.

To address the challenge of annotation, a number of unsupervised representation learning approaches have been proposed [4, 14, 19]. These approaches extract common visual representations from the input image without annotation. However, such methods do not perform well in gaze estimation, as gaze estimation differs significantly from common visual tasks. Common visual tasks like object classification and detection require representations of overall appearance of the subject. On the other hand, gaze is a direction vector that manifested as rotations of the subtle eye structure in the image. Efforts have been made to design unsupervised gaze representation learning methods.
We propose the Multi-View Dual-Encoder, a face-based un-supervised gaze representation learning framework under un-calibrated multi-view settings. The MV-DE is robust to free head movements.

- We uncouple gaze from other interference such as head pose by the proposed Dual-Encoder architecture and the multi-view gaze representation swapping strategy.
- Extensive experiments demonstrate the effectiveness of the MV-DE framework in unconstrained environments. Qualitative analyses prove that the extracted gaze representation is disentangled from head pose and appearance.

2. Related Work

2.1. Supervised Gaze Estimation

Appearance-based gaze estimation approaches aim to estimate gaze from eye or face appearance directly. Early methods estimate gaze directions from eye images [9, 28, 36]. Zhang et al. first propose to utilize full face images and outperform eye-image-based methods [37]. Since then, most CNN-based gaze estimation methods employ face images as input. Chen et al. propose to utilize dilated convolution for gaze estimation [5]. Cheng et al. propose to employ the Transformer architecture for gaze estimation [6]. Some recent methods utilize both face and eye images as input for better estimation performance [1, 8, 17, 22]. Above methods are trained in an end-to-end manner with gaze annotations. A number of gaze estimation datasets are published. These datasets are collected under different scenarios with various devices, including web cameras [11, 37], 360° cameras [21], high-resolution cameras [39], eye trackers [29] and mobile devices [22]. With sufficient labeled data, gaze feature disentanglement is achieved by using GAN [10] and Nerf [32].

2.2. Multi-view Gaze Estimation

Before the development of Deep Learning methods, most conventional model-based methods utilize multiple cameras for gaze estimation [16]. Model-based methods reconstruct 3D eyeball models and obtain the gaze direction based on the optical geometry of human eye structures [15]. Model-based methods achieve remarkable estimation accuracy. However, these methods have high requirements for the shooting angle of cameras. Thus, these methods are usually employed in the Head Mounted Devices, such as Meta Quest Pro and Microsoft Hololens, where the cameras are approximately stationary relative to the eye of users.

Multi-view settings have also drawn a lot of attention in CNN-based gaze estimation methods. A few multi-view gaze estimation datasets have been proposed. Zhang et al. propose the ETH-XGaze dataset, which employs 18 high-resolution cameras to capture face images with large head pose range [39]. Park et al. propose the EVE dataset [29]. They use 3 web cameras and a industrial camera to
capture user face images. Qin et al. propose to utilize 3D face alignment approaches to generate multiple rotated images from a given sample [31]. These rotated images can be regarded as samples from different virtual cameras. Based on these datasets, CNN-based multi-view gaze estimation approaches have been proposed. These methods integrate features from different views by concatenating [24] or self attention [7] to improve estimation accuracy.

2.3. Unsupervised Gaze Representation Learning

Unsupervised representation learning has always been a hot topic in Computer Vision community. Recently, Contrastive Learning approaches have achieved satisfying performance in common Computer Vision tasks. Contrastive Learning methods generate multiple views of a given sample by different data augmentation methods and constrain the model to extract similar representation from these views [4, 14]. He et al. propose the Masked Autoencoders, a self-supervised learner trained by simply reconstructing masked images. However, these methods are designed to extract common visual representations, which do not perform well in the gaze estimation task.

To learn gaze representations in an unsupervised manner, Yu et al. propose to utilize the gaze redirection task to extract two-dimensional representations that relate to gaze pitch and yaw angles [35]. Sun et al. propose the Cross-Encoder, which learns gaze representations by a latent-code-swapping mechanism on eye-consistent image pairs and gaze-similar pairs [33]. Gideon et al. further propose to adapt the Cross-Encoder architecture to the multi-view setting [12]. Above methods all employ eye images as their input, as there are less gaze-irrelevant visual contents. However, appearance of eyes changes dramatically as head pose increases, makes it difficult for these methods to handle samples with large head pose distribution. Our aim is to introduce an unsupervised gaze representation learning approach that employs face images as input, which is robust to unconstrained head movements.

3. Method

We propose the Multi-View Dual-Encoder, a face-based multi-view unsupervised gaze representation learning method. Our aim is to extract gaze representations from the input face images without using the gaze label. The main challenge of this task is to separate gaze from the other facial information, especially head pose. Since gaze and head pose are both physical directions, common data augmentations like rotation and flipping are not able to separate them.

To achieve our goal, the proposed MV-DE framework introduces two encoders: a Face Encoder and a Gaze Encoder. We first train the Face Encoder to extract general facial representation including head pose and appearance, while excluding gaze information. Then, we freeze the Face Encoder and train the Gaze Encoder to extract gaze representation which compensates for the missing gaze information. We constrain the Gaze Encoder to derive similar representation from different views within a given frame since the eyeball rotation is the same. In this way, we isolate gaze representation from other facial features without using the gaze label. In the following sections, we introduce the training pipeline of two encoders in detail.

3.1. Training of the Face Encoder

The target of the Face Encoder is to extract representation of general facial information except gaze. We design two training strategies to achieve this goal: eye masking and multi-task learning including head pose estimation and image reconstruction, as shown in Fig. 2.

![Figure 2. Training process of the Face Encoder. The proposed Face Encoder extracts face representations that contain general facial information while excluding gaze information. It is trained by two tasks: the head pose estimation task and the image reconstruction task.](image)

Given a training face image \( x_{i,j} \) where \( i \) is the frame index and \( j \) is the camera index, we mask the two eye areas by the average pixel value. We further add three random masks to prevent the Face Encoder from directly estimating head pose based on the position of eye masks. Given that input face images are normalized to \( 224 \times 224 \) pixels, we set the size of masks to \( 55 \times 33 \) according to the average size of eyes. Then, we input the masked face images to the Face Encoder for the proposed multi-task learning.

First, the Face Encoder extracts the face representation \( e_{i,j}^{\text{face}} \) from the masked face image. Then, we add a regression MLP and a decoder for the head pose estimation task and the image reconstruction task, respectively. In the head pose estimation task, the target of the Face Encoder is to estimate the 3D Euler Angles of the subject's head pose. We use \( L_1 \) Loss function for the head pose estimation task: \( L_{hp} = L_1(\hat{y}_{i,j}, y_{i,j}) \), where \( \hat{y}_{i,j} \) is the estimated head pose and \( y_{i,j} \) is the ground truth head pose. The purpose of the head pose estimation task is to ensure that the face representation \( e_{i,j}^{\text{face}} \) encodes head pose information.

In the image reconstruction task, the Face Encoder and the decoder are trained in an adversarial way. The target
Figure 3. Training process of the Gaze Encoder. The Gaze Encoder extracts gaze representations that represent the eyeball rotation, i.e., gaze direction within HCS. The Gaze Encoder is trained by the proposed multi-view gaze representation swapping strategy.

of the Face Encoder is to reconstruct the image with eye-masks. The target of the decoder is to reconstruct the original image from the face representation $e^{f}_{i,j}$, including the eye area. Both the Face Encoder and the decoder are constrained by the Image Reconstruction Loss $L_{re}$:

$$L_{re}(\hat{x}, x) = \alpha_1 L_2(\hat{x}, x) + \alpha_2 L_{pcpt}(\hat{x}, x),$$

where $\hat{x}$ is the reconstructed image, $x$ is the target image, $L_2$ is the Mean Squared Error (MSE) Loss function and $L_{pcpt}$ is the Perception Loss function. $\alpha_1, \alpha_2$ are the coefficients of loss functions. The purpose of the image reconstruction tasks is to encode general facial information in the face representation while excluding gaze information. Images constructed from the $e^{f}_{i,j}$ are basically the original face images without iris and pupil.

Overall, we train the Face Encoder by minimizing $L_{FE} = \beta_1 L_{hp} + L_{re}$, where $\beta_1$ is coefficient to balance two tasks. Once the training of the Face Encoder is completed, we freeze the parameters of the Face Encoder and proceed to train the Gaze Encoder via the multi-view constraint.

### 3.2. Training of the Gaze Encoder

The target of the Gaze Encoder is to extract gaze representation $e^{g}_{i,j}$ that represents the direction of gaze in the HCS without any gaze label. Since the face representation $e^{f}_{i,j}$ encodes general face information except gaze, we combine the gaze representation $e^{g}_{i,j}$ with $e^{f}_{i,j}$ to fully reconstruct the original image, so that $e^{f}_{i,j}$ compensates for the missing gaze information. The training strategy of the Gaze Encoder is shown in Fig. 3.

Given a training sample $x_{i,j}$, we extract the gaze representation from the original image and the face representation from the eye-masked image. Then, we input both representations to the decoder to reconstruct the original image. To reconstruct the original image, the Gaze Encoder has to capture gaze information which is missing in the face representation. Considering that the region occupied by two eyes represents only a small portion of the face image, we additionally compute the Image Reconstruction Loss for both eye regions:

$$\hat{x}_{i,j} = G(e^{f}_{i,j}, e^{g}_{i,j}),$$
$$L_{GE} = L_{re}(\hat{x}_{i,j}, x_{i,j}) + L_{re}(\hat{x}^{e}_{i,j}, x^{e}_{i,j}),$$

where $G(\cdot)$ is the decoder, $\hat{x}^{e}_{i,j}$ and $x^{e}_{i,j}$ are the two eye regions of the reconstructed image and the original image, respectively.

However, the gaze representation can also encode other facial information such as head pose during the training process. In an extreme case, the decoder might theoretically reconstruct the entire image solely based on the gaze representation, since the Gaze Encoder utilizes the full face image as input. We utilize the consistency of eyeball rotation in the multi-view settings to exclude gaze-irrelevant information. First, we randomly sample an image $x_{i,j}$ from another view within the same frame where $j' \neq j$. Since the eyeball rotations from both views are consistent, we swap the gaze representations of two views and reconstruct the eye regions of the original views. For a training image pair \{ $x_{i,j}, x_{i,j'}$ \}, the final loss function of the Gaze Encoder $L_{GE}$ is:

$$L_{GE} = \beta_2 L_{face}^{f} + \beta_3 L_{re}^{eyes} + \beta_4 L_{re}^{swap},$$

where $\beta_2, \beta_3, \beta_4$ are the coefficients, $L_{face}^{f}$ is the Image Reconstruction Loss of full face images from both view:

$$L_{face}^{f} = L_{re}(\hat{x}_{i,j}, x_{i,j}) + L_{re}(\hat{x}_{i,j'}, x_{i,j'}),$$

$L_{re}^{eyes}$ is the Image Reconstruction Loss of eye regions from both view:

$$L_{re}^{eyes} = L_{re}(\hat{x}^{e}_{i,j}, x^{e}_{i,j}) + L_{re}(\hat{x}^{e}_{i,j'}, x^{e}_{i,j'}).$$
\( \mathcal{L}_{\text{swap}} \) is the Image Reconstruction Loss of the generated eye regions after gaze representation swapping:

\[
\mathcal{L}_{\text{swap}} = \mathcal{L}_{\text{re}}(G(e_{i,j}^f, \epsilon_{i,j}^g)^e, x_{i,j}^e) + \mathcal{L}_{\text{re}}(G(e_{i,j}^f, \epsilon_{i,j}^g)^e, x_{i,j}^e').
\]  

At test time, the Gaze Encoder extracts gaze representations from the given single-view face images, similar to common supervised single-view gaze estimation methods. We rotate the estimated HCS gaze to CCS by the head pose label for evaluation.

### 3.3. Implementation Details

**Training Details:** The proposed method is implemented by PyTorch using two RTX 3090 GPU. We employ the Adam optimizer with a leaning rate of 10^{-3} for all the encoders and decoders. The Face Encoder is trained for 13 epochs with a batch size of 100. Learning rate is decayed by 0.2 every 4 epochs. The Gaze Encoder is trained for 5 epochs with a batch size of 50. Learning rate is decayed by 0.2 every 2 epochs. We use the data rectification method from [37] and histogram equalization to normalize the input face images. \((\alpha_1, \alpha_2)\) are set to \((10, 1)\) and \((10, 0.5)\) during the training of the Face Encoder and Gaze Encoder, respectively. \(\beta_3\) is set to 0.1 and \((\beta_2, \beta_3, \beta_4)\) is set to \((0.3, 0.35, 0.35)\). We use the first 6 convolutional layers of a ImageNet pretrained VGG-16 to calculate the Perceptual Loss.

**Network Architecture:** Both the Face Encoder and the Gaze Encoder use the ResNet-18 [18] as backbone. For the Face Encoder, we take the \((512 \times 7 \times 7)\) feature map after the last residual block as the Face Representation. The output channels of the last Linear Layer is set to 3 for head pose estimation. For the Gaze Encoder, the Linear layers is replaced by a \(1 \times 1\) Convolutional layer to compress the \((512 \times 7 \times 7)\) feature map to \((1 \times 7 \times 7)\) gaze representation. In the decoder, the channel of the gaze representation is expanded to 10 by a \(1 \times 1\) Convolutional layer and then concatenated with the face representation. The decoder also has four residual blocks with target channel sizes of \([256, 128, 63, 32]\). The feature maps are up-scaled to twice their size before each residual block. We upscale the feature map again and use two Convolutional layers to generate the final \((3 \times 224 \times 224)\) image. During the training of the Face Encoder, we set the gaze representation to zeros. Note that we train the decoder from scratch during the training of the Gaze Encoder, as the decoder has learned to ignore the 0 replaced gaze representation during the training of the Face Encoder.

### 4. Experiments

#### 4.1. Data Preparation

We conduct experiments on four different gaze estimation datasets: ETH-XGaze [39], MPII-NV [31], EVE [29] and EyeDiap [11]. We visualize the head pose and gaze distribution of each dataset in Fig. 4. The distribution of these datasets varies significantly. Below we introduce each dataset in detail.

**ETH-XGaze** (\( \mathcal{D}_E \)): A multi-view dataset collected by 18 high-resolution cameras. It provides very large head pose and gaze distributions. ETH-XGaze dataset contains 80 subjects, results in over 750,000 images in total. We divide the last 5 subjects as the labeled test set and the first 75 subjects as the multi-view unsupervised training set. We follow [25] to optimize the gaze and head pose label to ensure multi-view consistency.

**MPII-NV** (\( \mathcal{D}_M \)): A synthesized multi-view dataset. We follow [31] to reconstruct 3D faces from MPIIFaceGaze [37] dataset and rotate every 3D face to generate 18 different views, referring to the setting of ETH-XGaze. MPII-NV dataset contains 15 subjects and 359,984 images in total.

![Figure 4](image.png)

**Figure 4.** Head pose (top row) and gaze direction distributions (bottom row) of 4 different datasets. The EVE and the EyeDiap datasets only provide very limited head pose and gaze range with frontal faces.

![Figure 5](image.png)

**Figure 5.** Image reconstruction results of combining face representations with different gaze representations in the ETH-XGaze dataset.
Table 1. Gaze estimation error of few-shot experiments in degrees. **Bold** numbers are the best results and **underline** numbers are the second best results.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ETH-XGaze 50</th>
<th>ETH-XGaze 100</th>
<th>ETH-XGaze 200</th>
<th>ETH-XGaze Avg.</th>
<th>MPII-NV 50</th>
<th>MPII-NV 100</th>
<th>MPII-NV 200</th>
<th>MPII-NV Avg.</th>
<th>EVE 50</th>
<th>EVE 100</th>
<th>EVE 200</th>
<th>EVE Avg.</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calibration Num</td>
<td>50</td>
<td>100</td>
<td>200</td>
<td><strong>Avg.</strong></td>
<td>50</td>
<td>100</td>
<td>200</td>
<td><strong>Avg.</strong></td>
<td>50</td>
<td>100</td>
<td>200</td>
<td><strong>Avg.</strong></td>
<td><strong>Gain</strong></td>
</tr>
<tr>
<td>Cross-Encoder[33]</td>
<td>16.23</td>
<td>14.72</td>
<td>14.54</td>
<td>15.16</td>
<td>11.47</td>
<td>11.36</td>
<td>11.75</td>
<td>11.53</td>
<td><strong>8.07</strong></td>
<td><strong>7.81</strong></td>
<td><strong>7.48</strong></td>
<td><strong>7.79</strong></td>
<td><strong>11.49</strong></td>
</tr>
<tr>
<td>DE (ours w/o MV)</td>
<td>23.32</td>
<td>19.93</td>
<td>17.76</td>
<td>20.34</td>
<td>11.64</td>
<td>10.73</td>
<td>10.74</td>
<td>11.04</td>
<td>10.79</td>
<td>9.73</td>
<td>9.44</td>
<td>9.99</td>
<td>13.79</td>
</tr>
<tr>
<td>MV-DE (ours)</td>
<td><strong>8.66</strong></td>
<td><strong>8.08</strong></td>
<td><strong>7.77</strong></td>
<td><strong>8.17</strong></td>
<td><strong>7.28</strong></td>
<td><strong>6.52</strong></td>
<td><strong>6.39</strong></td>
<td><strong>6.73</strong></td>
<td><strong>9.18</strong></td>
<td><strong>8.6</strong></td>
<td><strong>8.68</strong></td>
<td><strong>8.82</strong></td>
<td><strong>7.91</strong></td>
</tr>
</tbody>
</table>

Gain

| ▼ 7.57 | ▼ 6.64 | ▼ 6.77 | ▼ 6.99 | ▼ 2.91 | ▼ 4.21 | ▼ 4.35 | ▼ 4.31 | ▼ 1.11 | ▼ 0.79 | ▼ 1.20 | ▼ 1.03 | ▼ 3.58 |

Figure 6. Gaze estimation error relative to the gaze yaw angle in the ETH-XGaze dataset. The proposed MV-DE performs significantly better when the gaze yaw angle is large.

We use the last subject as the labeled test set.

**EVE (D_1):** A multi-view dataset with 4 camera views. As shown in the Fig. 4, the EVE dataset only provides frontal faces with limited head pose and gaze distribution. We use the training and testing set split as defined in the original dataset and sample 3 images per second from the original videos, result in 527,896 images for unsupervised training and 64,464 images for testing.

**EyeDiap (D_2):** A single-view dataset with limited head pose and gaze distribution with 16,674 images in total. We only use the EyeDiap in the cross-domain experiments. Thus, we use the whole dataset for training or testing.

We follow [38] to normalize the face images. We employ Histogram Equalization and normalize the pixel values to \([-1, 1]\).

4.2. Evaluation of the Learned Representations

In this section, we validate the effectiveness of the learned face and gaze representations through image reconstruction. The propose MV-DE is supposed to decouple representations of the eyeball rotation from general facial information. In Fig. 5, we combine the face representation with different gaze representations and reconstruct the face image for validation. As shown in row 2, the decoder effectively reconstructs the input face image from the original face and gaze representation. When replacing the original gaze representation with gaze representation from another view, the reconstruction results are similar with the original one. It proves that the MV-DE successfully learned the multi-view consistency of the eyeball rotation. In the last row, the gaze representation is replaced by zero vectors. The head pose and appearance of the reconstructed images are almost identical with the original image, while the eyeball rotations are rather random. These results confirm that the MV-DE successfully separate eyeball rotation from general facial information.

4.3. Application of the MV-DE: Gaze Estimation

The learned gaze representation can be used for gaze estimation by adding a MLP head calibrated under the few-shot learning setting. The estimation error is also an evaluation of the learned gaze representation. In the MV-DE, the MLP takes the gaze representation and the head pose estimation as input. The estimated head pose is first encoded to a 30 dimensional embedding and then concatenated with the gaze representation through 3 linear layers with target dimensions of \((64, 64, 2)\). The MLP head is trained to estimate the HCS gaze. Then, we rotate the HCS gaze by head rotation to obtain CCS gaze. Sigmoid function is employed in the activation layer.

In Tab. 1, we compare the MV-DE with 3 SOTA unsupervised learning methods. BYOL [14] and SimCLR [4] are two SOTA contrastive learning methods. We modify them to adapt the multi-view gaze representation learning task. We employ face images from different camera views in the same frame instead of data augmentation methods to generate the positive pairs. Cross-Encoder [33] is a SOTA eye-based unsupervised gaze representation learning method. Gideon et al. also propose an eye-based method [12], but their method requires sample pairs with different eyeball rotation while keeping the head stable within a short video clip. Since such sample pairs are not available in \(D_E\) and \(D_M\), we exclude their method in comparison. We also remove the multi-view constraint in the MV-DE as an ablation study (named DE). Without the multi-view constraint, the Gaze Encoder is trained by the image reconstruc-
Table 2. Estimation error of different designs of the MLP head with 200 calibration samples.

| Dimension | MLP inputs & outputs | $D_E$ | $D_M$ | $D_V$
|-----------|----------------------|-------|-------|-------
| $d = 32$  | $e_g \rightarrow$ CCS Gaze | 24.34 | 42.82 | 9.83
|           | $e_g + HP \rightarrow$ CCS Gaze | 20.59 | 42.29 | 9.57
|           | $e_g \rightarrow$ HCS Gaze | 8.89  | 10.49 | 8.66
|           | $e_g + HP \rightarrow$ HCS Gaze | 8.63  | 7.26  | 8.60
| $d = 64$  | $e_g \rightarrow$ CCS Gaze | 22.27 | 41.70 | 9.98
|           | $e_g + HP \rightarrow$ CCS Gaze | 18.11 | 34.56 | 9.55
|           | $e_g \rightarrow$ HCS Gaze | 8.43  | 7.44  | 8.68
|           | $e_g + HP \rightarrow$ HCS Gaze | 7.84  | 6.39  | 8.67
| $d = 256$ | $e_g \rightarrow$ CCS Gaze | 20.97 | 40.99 | 10.32
|           | $e_g + HP \rightarrow$ CCS Gaze | 18.88 | 34.05 | 10.39
|           | $e_g \rightarrow$ HCS Gaze | 8.07  | 6.50  | 8.72
|           | $e_g + HP \rightarrow$ HCS Gaze | 7.59  | 5.71  | 8.87

Table 3. Cross-dataset gaze estimation errors of using the MV-DE as an unsupervised pretrain method. We first pretrain the model on the ETH-XGaze or the MPII-NV dataset unsupervised. Then, we use the labeled training dataset to calibrate the MLP head.

<table>
<thead>
<tr>
<th>Method</th>
<th>Unsupervised Pretrain</th>
<th>Train Dataset</th>
<th>Test Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ETH-Xgaze</td>
<td>MPII-NV</td>
<td>EyeDiap</td>
</tr>
<tr>
<td>Baseline</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MV-DE</td>
<td>ETH-XGaze</td>
<td>EVE</td>
<td></td>
</tr>
<tr>
<td>MV-DE</td>
<td>MPII-NV</td>
<td>EVE</td>
<td></td>
</tr>
<tr>
<td>MV-DE</td>
<td>ETH-XGaze</td>
<td>MPII-NV</td>
<td>EVE</td>
</tr>
<tr>
<td></td>
<td>31.27</td>
<td>32.52</td>
<td>12.55</td>
</tr>
<tr>
<td></td>
<td>19.67 • 37.10%</td>
<td>10.45 • 67.67%</td>
<td>11.36 • 9.47%</td>
</tr>
<tr>
<td></td>
<td>25.42 • 18.11%</td>
<td>10.33 • 68.23%</td>
<td>11.58 • 7.73%</td>
</tr>
<tr>
<td>Baseline</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MV-DE</td>
<td>ETH-XGaze</td>
<td>EyeDiap</td>
<td></td>
</tr>
<tr>
<td>MV-DE</td>
<td>MPII-NV</td>
<td>EyeDiap</td>
<td></td>
</tr>
<tr>
<td>MV-DE</td>
<td>ETH-XGaze</td>
<td>MPII-NV</td>
<td>EyeDiap</td>
</tr>
<tr>
<td></td>
<td>4.46</td>
<td>39.46</td>
<td>23.58</td>
</tr>
<tr>
<td></td>
<td>25.15 • 42.92%</td>
<td>12.11 • 60.31%</td>
<td>10.4 • 35.09%</td>
</tr>
<tr>
<td></td>
<td>25.58 • 41.94%</td>
<td>11.51 • 70.83%</td>
<td>10.58 • 35.19%</td>
</tr>
</tbody>
</table>

Overall, the average accuracy of the MV-DE outperforms other SOTA methods significantly. BYOL and SimCLR perform worse than the gaze-specialized Cross-Encoder and the MV-DE, proves that common visual representations are not suitable for the gaze estimation task.

Most importantly, the MV-DE outperforms the Cross-Encoder significantly in the ETH-XGaze and MPII-NV datasets, where the head pose distributions across a wide range. On the other hand, the MV-DE achieves comparable performance with the Cross-Encoder in the EVE dataset, because the EVE dataset only contains frontal face images with limited head pose range.

To further investigate the performance gap between the Cross-Encoder and the MV-DE, we visualize the gaze estimation error relative to the gaze yaw angle within the ETH-XGaze and MPII-NV datasets, where the head pose distributions across a wide range. The results prove that the MV-DE are more robust to large head poses.

Results from the last 2 rows demonstrate the importance of the proposed multi-view gaze representation swapping strategy. The multi-view constraint brought an improvement as large as 5.88°. The Dual-Encoder (DE) still achieves better overall performance than SOTA constrastive learning methods, primary attributes to the proposed dual-encoder architecture.

4.3.1 Ablation Study: Design of the MLP Head

The MLP head is responsible for estimating gaze from the learned gaze representations under the few-shot learning setting. In Tab. 2, we verify the performances of different MLP designs with 200 calibration samples. For example, [d = 64, $e_g + HP \rightarrow HCS$ Gaze] denotes that the MLP includes 64-dimensional hidden layers, takes gaze representations and head poses as the input vector, and predicts gaze directions within Head Coordinate System (HCS). Note that in the MV-DE, we rotate the estimated HCS gaze with head pose label physically.

Results of Tab. 2 lead to 3 conclusions: (1) The MLP performs significantly better on predicting HCS gaze, since the learned gaze representations represent eyeball rotation instead of the CCS Gaze. Theoretically, combination of $e_g + HP$ should be enough to predict the CCS gaze, but the estimation error is still huge, probably due to the limited number of calibration samples and MLP parameters. (2) Head pose seems helpful for predicting the HCS gaze. This observation is consistent with previous research [37] that the additional information of the full face images helps in the gaze estimation task. (3) 256-dimensional hidden layers achieve the best performance in the 200-shot settings. But larger number of parameters require more calibration samples, the performance of 256-dimensional hidden layers degrades in 50 and 100 shot calibration settings. Considering the balance between the number of parameters and required calibration samples, we choose $d = 64$ for the MV-DE.

4.3.2 Additional Experiments: MV-DE as an Unsupervised Pretrain Method

Another potential application of the MV-DE is to used as an unsupervised pretrain method, since it is more cost-effective to collect large number of unlabeled face images than labeled ones. In Tab. 3, we employ the MV-DE to pretrain the Gaze Encoder without gaze label in the ETH-XGaze or the MPII-NV datasets. Then, we freeze the Gaze Encoder and only train the MLP head in the EVE or the EyeDiap dataset with gaze label.

To establish a baseline, we directly train the Gaze Encoder and the regression MLP in an end-to-end manner with gaze label in the labeled training dataset, just like common supervised gaze estimation methods. The cross-dataset errors of the baseline method are extremely large on the ETH-XGaze and MPII-NV datasets. Then, we use the labeled training dataset to calibrate the MLP head.
4.4. Application of the MV-DE: Gaze Redirection

The learned representations of the MV-DE can also be used for the gaze redirection task. Results in Fig. 8 show that the MV-DE effectively redirects the gaze direction according to the HCS gaze of the reference images, while keeping the head pose and identity unchanged. Fig. 8 demonstrates that the learned gaze representation is subject-independent. The head pose and identity of the reconstructed images are controlled by the face representations, where the eyeball rotation is controlled by the gaze representations as expected.

4.4.1 Linearity Invariance of the Gaze Representation

Gaze, as a physical direction vector, conforms to the principles of additivity. Ideally, well-learned gaze representations should possess similar characteristics. In Fig. 9, we combine gaze representations from different samples linearly with different ratios to reconstruct face images. The sphere in the bottom-left corner of reconstructed images indicates the target eyeball rotation angle, which corresponds to the linear combination of eyeball rotations from the HCS gaze source images. Results demonstrate that the learned gaze representation satisfies Linearity Invariance, i.e., the eyeball rotation of reconstructed images undergo the same linear combination as the input gaze representations. Experiments in Fig. 9 prove that the MV-DE successfully captures and retains part of the physical properties of gaze during the unsupervised learning process.

5. Conclusion

In this paper, we present the Multi-View Dual-Encoder (MV-DE), an unsupervised gaze representation learning framework based on multi-view face images. We propose the Dual-Encoder architecture and the multi-view gaze representation swapping strategy to learn gaze representations that are separated from general facial information without gaze label. Experiments show that the learned gaze representations can be used for downstream tasks like gaze estimation and gaze redirection. Gaze estimation results show that the quality of learned gaze representations from the MV-DE are significantly better than other SOTA methods with unconstrained head movements.


