Learning to Detect Head Movement in Unconstrained Remote Gaze Estimation in the Wild

Zhecan Wang*[1,2], Jian Zhao*†[3], Cheng Lu*‡[2], Han Huang§[2], Fan Yang¶[2], Lianji Li‖[2], and Yandong Guo**[2]

[1]Columbia University, [2]XPENG Motors, [3]Institute of North Electronic Equipment, Beijing, China

Abstract

Unconstrained remote gaze estimation remains challenging mostly due to its vulnerability to the large variability in head-pose. Prior solutions struggle to maintain reliable accuracy in unconstrained remote gaze tracking. Among them, appearance-based solutions demonstrate tremendous potential in improving gaze accuracy. However, existing works still suffer from head movement and are not robust enough to handle real-world scenarios. Especially most of them study gaze estimation under controlled scenarios where the collected datasets often cover limited ranges of both head-pose and gaze which introduces further bias. In this paper, we propose novel end-to-end appearance-based gaze estimation methods that could more robustly incorporate different levels of head-pose representations into gaze estimation. Our method could generalize to real-world scenarios with low image quality, different lightings and scenarios where direct head-pose information is not available. To better demonstrate the advantage of our methods, we further propose a new benchmark dataset with the most rich distribution of head-gaze combination reflecting real-world scenarios. Extensive evaluations on several public datasets and our own dataset demonstrate that our method consistently outperforms the state-of-the-art by a significant margin.

1. Introduction

Unconstrained remote gaze estimation has many important applications [24, 33, 11, 35, 1, 5] mostly around **H**uman **C**omputer **I**nteraction (**HCI**) [18, 26, 43]. A variety of existing methods [46, 14, 31, 39] could achieve very high accuracy in detecting gaze directions under controlled settings.

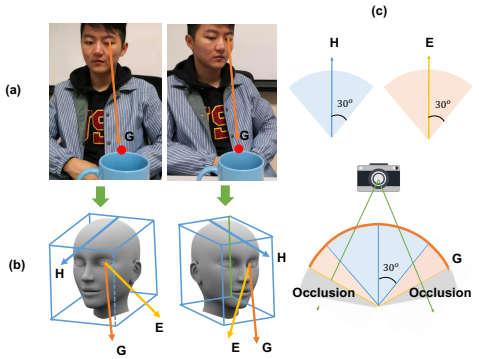

Figure 1: Effect of head movement on gaze. H represents head-pose vector, E represents eye-ball vector and G represents gaze vector. In (a), without head-pose, both poses map to the same gaze ground-truth respective to camera causing confusion. Even though the gaze vector, G, relative to camera coordinate stays the same, both head-pose vector, H, and eye-ball vector, E, change. However, with head-pose, it is easier to learn the difference and more accurate mapping function to estimate gaze direction. Head movement would also affect gaze distribution [34]. Since eye-ball vector rotates around a given head-pose vector, a function of the observed head-pose [34] is normally the mean of the gaze distribution. Further, as illustrated in (b), assuming the ranges of head movement and eyeball movement are up to 60 degrees. Thus, the head-pose could cover up to 60° in total. However, based on head-pose, gaze could cover up to 120° in total. In addition, if the head moves to the edge of its distribution, the eye movement may have occlusions against the camera. These occlusions would cause differences in gaze's ranges of distributions corresponding to head-pose.

However, existing methods [22, 14, 31] still suffer from problems like: inaccuracy under real-world conditions, need of complex settings to adapt to free-head movement, low image quality [53], offset from personal calibration, *etc*. Among them, head movement perhaps is the most crucial factor that significantly affects unconstrained remote gaze estimation of the following reasons, 1) any gaze vector re-

*olinzhecanwang@gmail.com
†zhaojian90@u.nus.edu, https://zhaoj9014.github.io/
‡lvcheng27@hotmail.com
§huangh@xiaopeng.com
¶fyang@temple.edu
‖lilj2@xiaopeng.com
**yandong.guo@live.com, Corresponding author

lated to a fixed camera coordinate depends on both eye vector (visual axis of an eyeball [10]) and head-pose vector, 2), as illustrated in Fig. 1, head-pose also strongly affects gaze distribution including both mean and range [34], 3) head-pose would change eye appearance [11]. The difference in head-pose would cause geometric deformation. Eye regions like pupil, iris, sclera, *etc*, would be occluded to different extents [34]. Because this deformation is holistic throughout the face, it is too diminutive in a local eye region for appearance-based methods to detect and track especially without personal calibration. With this understanding, we believe gaze estimation could be more robust to the change of eye appearance caused by head movement by incorporating head-pose information. In this work, we introduce two ways of incorporating head-pose into gaze estimation to achieve better accuracy.

Among unconstrained remote gaze estimation methods, appearance-based methods recently become popular due to their general applicability to multiple scenarios [52, 31, 14, 17, 39, 53]. However, they are also not sensitive enough to free-head movement especially when eye ball's relative position to camera coordinate is fixed, as in Fig. 1. Furthermore, they are trained and evaluated on public datasets mostly collected under controlled scenarios with very limited illuminations, subject identities, backgrounds, *etc* [8, 14, 38, 48, 40, 31, 9, 42, 23, 53, 38]. Most importantly, these datasets lack rich distribution of head-pose. Some of their sampling ranges are even discrete. Due to these problems, these datasets bear the risk of bias and could not generalize to other real-world scenarios, *e.g.* in-car scenarios under sunlight.

We compensate the confusion caused by head movement in gaze estimation by introducing two ways of incorporating head-pose. Our work focuses on proposing a system to incorporate head-pose in two different scenarios. First, when direct head-pose information, *i.e.* facial image and head-pose vector, is available, we propose **H**ead-pose-aware **G**aze **D**etector (**HGD**), an appearance-based method that leverages head-pose and gaze in an end-to-end structure. Different from previous works like [34, 52], our method merges head and gaze information more properly in different levels of representations including hidden feature level, training task level, and model level. On each level, these representations are merged with similar spatial dimensions and information complexities. HGD outperforms the state-of-the-art in both public datasets and our dataset. Furthermore, in some scenarios (datasets) where direct head-pose information is not preserved, we additionally propose a side method, **HGD-no-H**ead-**P**ose (**HGD-noHP**), that could also incorporate head-pose into gaze estimation by extracting head-pose information from eye deformations. In order to evaluate our methods better on a benchmark closely reflecting real-world scenarios, we further collect our own dataset, *i.e.* In-car Gaze dataset. In this dataset, we collect data from both head and eye movement over much larger continuous ranges compared with existing datasets.

Our contributions are summarized as follows.

- We propose an end-to-end method, HGD and one side method, HGD-noHP, for better incorporating head-pose in gaze estimation in the wild.
- For better evaluating our frameworks, we collect a large-scale benchmark with richer head-gaze distribution better reflecting real world scenarios.
- Comprehensive evaluations on the In-car Gaze dataset proposed in this work and other existing datasets verify the superiority of our frameworks on gaze estimation in the wild over the state-of-the-art.

2. Related Work

Recent remote gaze estimation methods focus more on head-free gaze estimation by incorporating head-pose information [46, 8, 45, 41, 21]. They could be divided into two main categories, *i.e.* appearance-based and model-based methods.

Model-based methods often use the geometric prior to regularize models for gaze estimation. They are previously widely explored for good accuracy and ability to handle free-head movement by using multiple light sources or cameras under controlled settings [15, 19, 28, 2, 56]. They could be divided into two parts, **P**upil **C**enter **C**orneal **R**eflection methods (**PCCR**) [11, 10] and non-PCCR methods depending on if using external light sources or not. PCCR methods could be precise in controlled scenarios but impractical for real-world scenarios. Non-PCCR methods include 3D model-based methods [25, 10, 13] and 2D shape-based methods [4, 54]. 3D model-based methods and 2D shape-based methods directly infer gaze from observed eye shapes, such as pupil center or iris edges. If applied to real-world scenarios, model-based methods could not easily adapt to free-head movement, low image quality, different lightings or subjects without extra calibration. This complexity limits them from being applied to more general environments.

Appearance-based methods directly use eye images as input and can, therefore, work with low-resolution images. Because they are typically data-driven, they could leverage large amounts of head-pose independent training data to generalize to arbitrary users without extra setup or calibration. Current works using monocular cameras become more attractive given its generality [55]. Even though existing appearance-based methods do include head-pose information in the pipelines but they do not incorporate it properly. In existing methods like [3, 8, 52, 34], the measured 3D head-pose vector is directly inserted into the second last **F**ully **C**onnected (**FC**) layer. This direct concate-

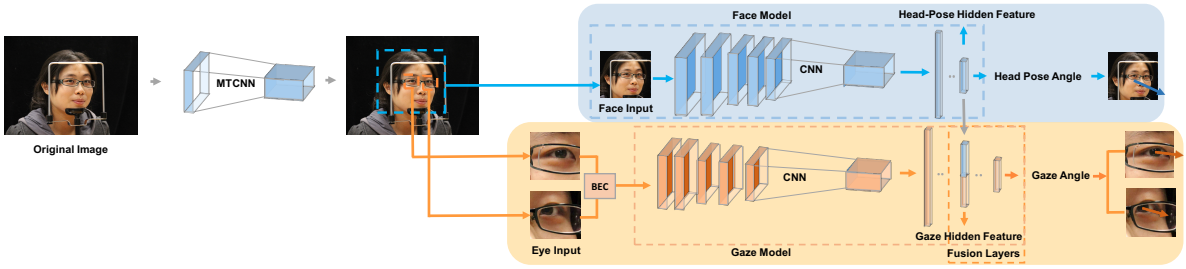

Figure 2: Structure of HGD. In this framework, head-pose and gaze are merged and have a balance in input level (we use both face and eye images as inputs in the input level for models), hidden feature level (concatenation between two hidden features), model level (parallel relationship between head model and gaze model) and task level (parallel relationship between head-pose task and gaze task). The blue part represents the training and testing on head-pose information and the red part on gaze information (we enhance the resolution of input images here for demonstration purpose). Best viewed in color.

nation would be difficult for the last FC layer to learn since the 3D head-pose vector is very different from the learned hidden features in terms of spatial dimension and embedded information complexity. In other ways of incorporating head-pose, [17] proves that a **C**onvolutional **N**eural **N**etwork (**CNN**) that takes multi-region inputs, *i.e.* eyes, faces and face grids, can improve gaze estimation. These regions, mainly the full face, could better encode head-pose, geometric structure of head and illumination across larger areas than those available in the eye region. Thus, from experiments in [17, 39], we observe that full-face-based CNNs are more accurate for gaze estimation than eye-image-based CNNs. However, they are limited in applicability as the entire face may not be available in all scenarios [14]. Furthermore, the method proposed in [17] may be limited to 2D-screen scenarios and the full face-based method in [39], only using full face as input, may be more vulnerable to low image quality where eye regions could be more blurry. Unlike [52, 34], our methods merge head-pose and gaze when they are in similar levels of representations, *e.g.* merging between hidden feature vectors, parallel learning tasks, *etc*. Different from [17, 39], our methods are also not limited to 2D screens, less vulnerable to low image quality and could better generalize to different scenarios.

3. Gaze Dataset

Even though many public gaze datasets are already available [8, 14, 38, 48, 40, 9, 42, 23, 53, 38, 17], many of them [23, 48, 38] are collected in controlled laboratory settings and have limitations in scales, subjects, ranges of sampling, *etc* [40, 9, 52, 14]. These limitations would cause problems like lack of variation for subject appearances, head-pose, gaze, *etc* [23, 42, 48, 38], and further prevent appearance-based methods from better generalization [46]. Thus, we collect our benchmark, In-car Gaze, closer to real-world scenarios to train more generalized appearance-based methods and more clearly demonstrate the advantage of our frameworks.

In-car Gaze not only has the largest continuous ranges of sampling for gaze and head-pose but also has one of the largest scales in frames. Many of the datasets like [23, 42] underplay the collection of head-pose information, *e.g.* most of them do not store facial images but only eye images. This causes an imbalance in the distribution, stored data format and quantity between head-pose and gaze information. We overcome this by focusing on the collection of both head-pose and gaze. Besides, most datasets [40, 52, 8] are recorded under controlled scenarios having limited participants [40, 9, 52, 14] and environment settings like illumination conditions and backgrounds. Differently, we invite 1000 participants with diverse facial appearances. Furthermore, our dataset is the only dataset that labels both left eye and right eye on the same face respectively with two different gaze ground-truths and has multi-camera views per shot (supplementary, Sec. In-car Gaze Dataset).

Our work do not solely focus on car driving scenarios. Different from existing car gaze solutions and datasets [29, 36, 47, 27, 44], our frameworks and datasets focus more on improving general gaze estimation by incorporating head-pose information. The flexibility of our solutions and detailed labelling of In-car Gaze dataset could generalize to other daily life scenarios.

4. Proposed Method

In the following sections, we introduce methods to incorporate head-pose into gaze estimation in two different scenarios: head-pose learned from human face when direct head-pose information, *i.e.* both face images and head-pose labels, is available and head-pose learned from eye deformations when direct head-pose information is not available [11]. When merging, we consistently unify head-pose and gaze representations in a similar level of spatial dimension and embedded information complexity. We believe this intuitive strategy would help our models better incorporate head-pose to reach higher gaze estimation accuracy. Furthermore, realizing that the distance between two pupils causing asymmetry, we find that **B**oth **E**yes concatenated on the **C**hannel (**BEC**) level could help achieve the best accuracy and efficiency compared with single eye method and else, referring to **Component Analysis**. Thus in both of our frameworks, eye images are pre-processed in **BEC** method on our dataset but in the fashion of single eye in

public datasets when paring information is not available.

4.1. Head-Pose Learned from Human Face

Eye image, the direct local information, is important for gaze prediction. However, for appearance-based methods, the change of eye appearances from head movement may be too diminutive to detect solely from this local information. Thus, the change of eye appearances caused by head-pose would cause confusion for the regressor. To solve this, we introduce extra global information by bringing in full face information. This is because geometric deformation caused by head movement will be more distinctly expressed in the scale of full face. We further formulate this learning problem as a task of learning a transformation function, $F_{transform}$, from eye, X_{eye}, and face, X_{face}, to gaze prediction, g_w, as in Eqn. 1. With this intuition, in scenarios, *e.g.*, our collected dataset, where both facial images and head-pose labels are available, we propose our main method, HGD, as illustrated in Fig. 2. The original image is passed through a MTCNN face detector [51] to produce face image and eye images based on detected landmarks. Then the remaining framework learns both head-pose (as the blue part) and gaze (as the red part) from these face and eye images. [39] uses spatial weights to focus on the edges, the geometric layout of face besides eye regions. Different from that, in our method, this weighting could be implicitly learned through the head-pose prediction task from face image. We use a simple ResNet-34 [12] structure as the face model to learn head-pose directly from face image, as the top part in Fig. 2. In this setting, head-pose information is implicitly embedded in the geometric structure of the provided face image. The face model outputs a 64×1 feature vector from the second last FC layer and a 3D head-pose angle (yaw, pitch). This part is formulated as the first equation in Eqn. 2. We then also have a ResNet-34 [12] as the gaze model to produce a gaze hidden feature (64×1) at its FC-3 layer, formulated as the second equation in Eqn. 2. The gaze model concatenates the shared head-pose feature with its gaze feature and then outputs to the fusion layers, following FC layers, to predict gaze. This part is formulated as the last part in Eqn. 2, also illustrated in Fig. 2. From this framework, the gaze model not only learns head-gaze relationship from the back-propagation from both training tasks but also from the concatenated features. This end-to-end schema allows the model more easier handle low image quality and adapt to different scenarios.

For implementation, depending on the distribution of head-gaze distribution in different datasets, we have two training strategies for this structure.

Multi-task, Implicit Learning: Public datasets, as Columbia Gaze [38] or MPII Gaze [52] datasets shown in Fig. 5, usually have insufficient combination of head-pose and gaze due to insufficient collection of head-pose. Therefore, it would be easier for the model to learn the head-gaze relationship even though these datasets may not truly reflect the real-world scenarios. In this case, we train the face model and the gaze model jointly on two parallel tasks, one head-pose loss and one gaze loss, as in Eqn. 3. The learning of face model and the designated loss function would force the gaze model to learn the relationship between gaze and head-pose thus helping gaze prediction. The backpropagation from two losses would simultaneously constrain face model and gaze model mutually. We set the model to mainly learn to predict gaze and assist this learning with an ancillary head-pose task. Purposely, we multiply the head-pose loss with a weakening factor so as to strengthen the gaze learning during training. Because of the intrinsic characteristics of deep learning, we could not fully supervise the whole learning process during multi-task learning and ensure that the gaze model could learn head-gaze relationship properly in every step. Consequently, we call this implicit learning strategy.

Multi-stage, Explicit Learning: During training in our dataset, we realize that the losses of both head-pose and gaze could not converge jointly as well as we experience in public datasets like Columbia Gaze [38]. This may be due to facts that in real-world scenarios as in our dataset, the distribution of head-gaze is very dispersed, as shown in the right of Fig. 5. Also, different from most public dataset collected in controlled laboratory scenarios, our dataset is collected in daily scenarios and the labelling could be rough. Thus it would be more difficult for the model to learn the relationship between head-pose and gaze jointly online. In this case, we sacrifice computation efficiency to conduct a multi-stage training strategy. We first only train the face model with the head-pose loss until it converges well. Then we freeze the face model and use its inferenced output, *i.e.* head-pose hidden features, to feed the gaze model for gaze prediction. Under this strategy, we are able to secure the stable performance of both models with less internal constraints during training. In this setting, we specifically train the face model on head-pose prediction and it back-propagates only on its own and so as gaze model. Comparatively, we call this strategy explicit learning since we separate the learning processes explicitly upon two tasks. As shown in Tab. 3 and Tab. 1, HGD achieves the best accuracy in our dataset and public datasets where head-pose information is well preserved.

$$g_w = F_{transform}(X_{eye}, X_{face}), \tag{1}$$

$$F_{transform} = \begin{cases} V_{face} = CNN_{face}(X_{face}, W_{face}), \\ V_{eye} = CNN_{eye}(X_{eye}, W_{eye}), \\ g_w = F_{fusion}(V_{face}, V_{eye}), \end{cases} \tag{2}$$

$$Loss_{batch} = \sum_{n=1}^{M} (Loss_{gaze}^{(n)}(g_w, l_g) + \beta \cdot Loss_{head}^{(n)}(h, l_h)). \tag{3}$$

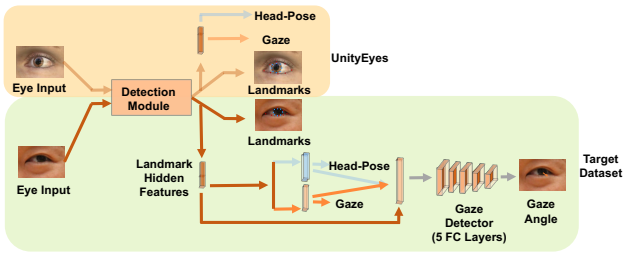

Figure 3: Structure of HGD-noHP. The yellow area denotes that the landmark detector is trained on photo-realistic synthetic data from UnityEyes [49]. The green area denotes the inference of the landmark detector and the training of gaze detector on target dataset.

Where $F_{transform}$ indicates the predict function from eye and face to gaze, g_w represents the predicted gaze, CNN_{face} and CNN_{eye} represent head-pose and gaze model respectively, V_{face} and V_{eye} are respective hidden features, W_{face} and W_{eye} represent parameters in both models respectively, F_{fusion} means the fusion layers, M is the batch size, β is the weakening factor, both l_g and l_h are the ground-truth labels and h means the head-pose.

4.2. Head-pose Learned from Eye Deformation

In some of the public datasets, no head-pose information is provided. In real-life scenarios, sometimes only eye images would be provided for remote gaze estimation so our model needs to be very robust to the offset from free-head movement while maintaining accuracy. As mentioned previously, based on works from [11, 34], we know that head movement would change eye appearances, as further demonstrated in Fig. 4. With investigation, we realize that head-pose can also be approximated reversely from eye appearances solely, mainly eye features, *e.g.* shapes of pupils and iris.

After inspired by model-based gaze estimation algorithms [31], we designed a new appearance-based algorithm, HGD-noHP, that focuses on predicting gaze from eye's deformations. Eyeball movement would mainly force the movements of pupils and iris regions causing deformations respective to camera. However head movement would not only cause the deformations from pupils and iris but also the overall structure of eye regions including eyelids, *etc*. It is indeed hard to differentiate between these two kinds of causation relationships explicitly. Thus, instead of directly learning attention maps or gazemaps as in [30] to mask out specific regions like iris or eyeballs, we utilize labeled data from UnityEyes [1] [49] to learn those two mappings from two target losses, *i.e.* head-pose and gaze losses. We believe this implicit learning could best utilize the strength of learning based methods.

[31] trains a tremendously large hourglass model on synthetic data to predict eye's landmarks and has a model-based framework followed to estimate the gaze based on these predicted landmarks. Differently, we use a much simpler model, ResNet-34 as the detection module to achieve better computation efficiency, shown in Fig. 3. Its main task is to serve as the backbone of a landmark detector to predict 16 landmarks of eye's interior margins and iris (8 landmarks for each category, 32 units in total). We first train the landmark detector on synthetic eye images from UnityEyes. Then we freeze the landmark detector and extract the inferenced hidden features out from the second last FC layer of 200 units. Further, we feed those hidden features to two additional modules, gaze module and head-pose module. Each of them consists of 5 FC layers to train to predict gaze and head-pose respectively on UnityEyes. Those learning tasks would train two modules to learn the important mappings from landmark hidden features to gaze and head-pose.

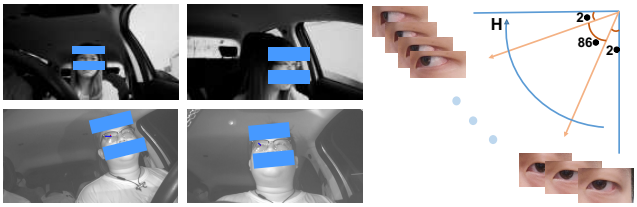

Figure 4: Example of multi-camera views in car. We collect data through 4 cameras in different perspectives to allow models more generally learn gaze estimation in real-world scenarios. The 1st row shows images collected in daytime scenarios and the 2nd row shows images collected in nighttime. Deformation of eye appearance respective to camera view due to change of head-pose without eyeball movement. The change of appearance may not be clear when head movement is small but obvious when large.

Different from [31] only directly using the predicted landmarks, on the target dataset, *e.g.* Columbia dataset, we first extract inferenced features from landmark detector. Based on that, we also extract inferenced features of 200 units from both gaze and head-pose modules. We concatenate those three hidden features as input to train a final gaze model of 5 FC layers. We believe the concatenated hidden features have richer information about deformations of eye corresponding to both head-pose and gaze than just landmarks. Furthermore, we also use the SimGAN [37] trained on the target dataset to help improve the synthetic data by adding more realistic elements (supplementary, Sec. Synthetic Eyes from UnityEyes Improved by SimGAN). This would compensate the accuracy loss due to the decreased capacity of landmark detector. By learning landmark features in the first stage, we add prior knowledge to guide the first part of this framework and believe it would learn the essential geometric features of eyes and the mappings from those geometric deformations to both gaze and head-pose. The training is conducted on photo-realistic synthetic data from UnityEyes improved by SimGAN since labeling eyes' landmarks to the details of interior margins and iris could

[1] https://www.cl.cam.ac.uk/research/rainbow/projects/unityeyes/tutorial.html

be very ambiguous and tedious. We further demonstrate that this side framework may not achieve better accuracy than our main proposed method but still outperforms the state-of-the-art in our dataset as shown in Tab. 3 and other existing datasets like Columbia [38] and MPIIGaze [52], as in Tab. 1 and Tab. 2.

5. Experiments

In this section, we first list our implementation details and two evaluation metrics. We then thoroughly analyze the importance of different parts in our algorithms through component analysis. Lastly, we not only evaluate our algorithms with the state-of-the-art on public datasets but also on our own benchmark.

5.1. Implementation Details

Data Preprocessing: In our dataset, images are stored in grayscale and sometimes have overexposure due to various lightings. For alleviating the effect of overexposure, we use **M**ultilevel **H**istogram of **O**riented **G**radients (**MHoG**) [6] + **L**inear **D**iscriminant **A**nalysis (**LDA**) which is invariant to various illuminations to certain extent, as suggested in [14]. We use MHoG + LDA to first extract features from images and then concatenate it with the original images to feed into the framework (supplementary, Sec. 10 Data Preprocessing). This preprocessing could help improve testing accuracy in our dataset, as in Tab. 4.

Training: We leverage Pytorch [32] as the implementation environment and our experiments are conducted on a single NVIDIA GPU with 16 GB memory. Our frameworks are trained for 100 epochs with batch size of 64. The input images are set to be 224×224. The starting learning rate is set to 0.0001 and decays by 0.1 every 30 epochs. Wing loss [7] is adopted in our methods. For HGD, after many experiments, the weakening factor, β is empirically set to 0.3 during online multi-task training. In HGD-noHP, gaze and head-pose module each consists of 5 FC layers of sizes: 200, 200, 100, 50 and 2. The gaze model consist of 5 FC layers of sizes: 600, 300, 100, 32 and 2 to predict gaze. We use UnityEyes to generate 100,000 synthetic images (90,000 for training, 10,000 for testing).

Evaluation Metrics: Different papers use their own evaluation metrics as in [16]. For sharing the same evaulation standard, we consistently use two methods in our work (supplementary, Sec. Evaluation Metrics). **V**ector **E**rror **M**etric (**VEM**) calculates the 3D angle difference between the predicted, P, and the labeled 3D vector, R, as in Eqn. 4 and Eqn. 5. We also use **A**ngle **E**rror **M**etric (**AEM**) to calculate the real difference in angular values between the predicted angle, (Θ_p, α_p), and labeled angle, (Θ_r, α_r), and ensure their real values are not far off, as in Eqn. 6.

$$P = T(\Theta_p, \alpha_p), R = T(\Theta_r, \alpha_r), \tag{4}$$

$$D_{VEM} = \arccos(P \cdot R), \tag{5}$$

$$D_{AEM} = \frac{1}{2n} \sum |\Theta_p - \Theta_r| + |\alpha_p - \alpha_r| . \tag{6}$$

Where T represents the transform function from 3D angles to vectors, P represents the predicted angle, R represents the labeled angle and n represents the number of samples in test data.

5.2. Evaluation on Public Dataset

5.2.1 Evaluation of Gaze Estimation with Direct Head-pose Information

To better demonstrate the advantage of our algorithms over the state-of-the-art, we further evaluate our algorithms over three public datasets.

Backbone	Framework	Columbia [38]		MPII [52]		GazeCapture [17]
		AEM	VEM	AEM	VEM	MSE
ResNet-34 [12]	HGD - Exp	0.84	**1.32**	NA	NA	**2.10**
	HGD - Imp	**0.82**	1.35	NA	NA	NA
	HGD-noHP	1.94	3.32	**4.02**	**5.33**	3.39
	MPIIGaze [52]	5.42	8.02	4.41	6.38	6.93
Lenet [20]	HGD - Exp	1.52	2.49	NA	NA	2.49
	HGD - Imp	1.59	2.41	NA	NA	NA
	HGD-noHP	2.34	3.45	4.31	5.52	3.92
	MPIIGaze [52]	5.32	8.26	4.51	6.43	8.03
	iTracker [17]	4.1	7.32	NA	NA	2.13
	RedFTAdap [50]	3.54	NA	5.35	NA	NA
	PictorialGaze [30]	3.8	NA	4.5	NA	NA
	Bayes-adversarial [46]	NA	NA	4.3	NA	NA

Table 1: Comparison of our algorithms with the state-of-the-art on public datasets (cross-subject). Eye image input is pre-processed in the fashion of single eye per unit.

As in Tab. 1, in all three public datasets, our frameworks could outperform the state-of-the-art. For a fair comparison, we also replace the backbone of our frameworks with Lenet [20] and they still achieve better accuracy than the state-of-the-art. MPII Gaze dataset [52] (not MPIIFaceGaze [39]) does not provide facial images and is collected in front of laptops causing limited distributions of head-gaze combination. However, HGD-noHP could still take benefit from incorporating head-pose related information to outperform or achieve a comparable accuracy against the state-of-the-art. Even though GazeCapture [17] is collected using phones or tablets and has a smaller distribution of head-gaze, our frameworks could still generalize on it. With this constraint, our frameworks may not be able to significantly demonstrate its advantage in incorporating head-pose information for gaze estimation. However, they could still achieve better accuracy against the state-of-the-art.

5.2.2 Evaluation of Gaze Estimation without Direct Head-pose Information

As mentioned earlier, in scenarios where direct head-pose information is not available through vector format or facial images, we could infer head-pose information through geometric deformations from eye. Our HGD-noHP framework focuses on learning eye features first and then transfer to gaze prediction. In Tab. 2, HGD-noHP outperforms the state-of-the-art by a significant margin with head-pose information removed purposely. This signifies the strong relationship between eye features and head-pose.

		Columbia [38]	UnityEyes [49]
HGD-noHP	AEM	**1.94**	**2.34**
	VEM	**3.32**	**3.45**
HGD-noHP w/o SimGAN [37]	AEM	2.51	NA
	VEM	4.17	NA
ResNet-34 [12]	AEM	3.29	4.24
	VEM	5.39	5.92
MPIIGaze [52]	AEM	5.4	5.12
	VEM	8.42	7.98
M-3D Gaze [55]	AEM	4.09	4.87
	VEM	6.2	5.67

Table 2: Comparison of HGD-noHP and other algorithms on public dataset when head-pose information is removed purposely (degree).

Method	AEM	VEM
HGD - Imp	3.69	5.17
HGD - Exp w/ BEC	**1.79**	**2.87**
HGD - Exp	2.53	3.67
HGD-noHP w/ BEC	2.94	4.6
HGD-noHP	3.21	4.97
iTracker [17]	5.61	8.64
iTracker with ResNet-34	5.56	8.07
MPIIGaze [52]	4.49	6.61
MPIIGaze with ResNet-34	3.71	5.44
M-3D Gaze [55] with ResNet-34	5.7	9.57

Table 3: Comparison of different head-gaze merging algorithms on our dataset (degree).

5.3. Evaluation on the Real-World In-car Gaze Dataset

During driving, the driver has a relatively broader range for head-pose and gaze among daily life activities, so we select driving as our base scenarios for data collection. As demonstrated in Fig. 5 (more detailed comparison in Tab. 1 of supplementary), In-car Gaze have the largest continuous sampling ranges of head-pose and gaze compared with existing datasets. 1,000 participants are invited from all different kinds of age groups and body traits to ensure the diversity. The collection is conducted inside a car with window and sunroof glasses open sitting outdoors throughout daylight and night to imitate the real-life daily scenarios. For designing a robust system, participants are also asked to wear a variety of different attires including sunglasses, glasses, hats, *etc*. Different from most, we also preserve facial images and label the gaze ground-truths for both left eye and right eye independently from the same face. Last but not least, 400 images are captured for each participant and a large scale of 400,000 frames are stored. 4 near infrared cameras (better visibility, less noise at night than RGB cameras) are set up inside the car in different positions toward the driver. During collection, our machine navigates a laser pointer point to the front within a prefixed grid. For each point, 4 photos are produced from 4 sync cameras, as in Fig. 4. In our dataset, we also store 9 facial landmarks, eye patches, face patches, recovered gaze ground-truths of both left and right eyes, and head-pose vectors.

We, in depth, compare our methods of head-gaze merging with the state-of-the-art. In order to evaluate the algorithms fairly, we also replace the backbone of iTracker

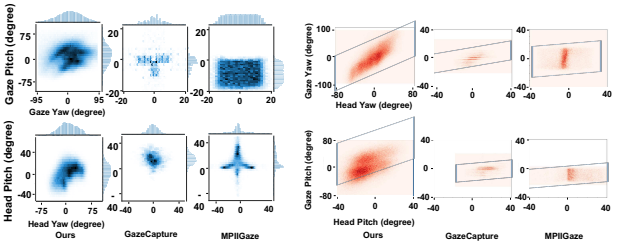

Figure 5: Comparison of distribution of head-pose and gaze across 3 datasets. On the left image, the first row shows the distribution of gaze and the second one shows the distribution of head-pose. The right image shows the distribution of head-gaze combination. The first row demonstrates the distribution in yaw direction and the second one in pitch. Our dataset has the biggest sampling ranges for both head-pose and gaze in both directions.

[17], MPIIGaze [52] and M-3D Gaze [55] frameworks with ResNet-34 [12]. As in Tab. 3, all of our presented algorithms could achieve better accuracy than the state-of-the-art with a significant margin. By merging head-pose representation, HGD with explicit learning method achieves the best accuracy. HGD-noHP, due to their limit in head-pose representation, achieve slightly worse accuracy. The original iTracker framework [17] takes left eyes, right eyes, faces and face grids as inputs for the gaze estimation task with 2D on-screen settings. However, in more general real-life settings, face grid may not be necessarily related with gaze estimation but causes more noises. Furthermore, MPIIGaze [52] and M-3D Gaze [55] do not merge head-pose and gaze as comprehensive as our frameworks, either, thus having accuracy drop in their corresponding results.

Classification: The practical use of this work is to assist to detects driver's attention. Thus, it may not be necessary to fully determine the exact angle where the driver is looking. In this case, we split the frontal space of the driver into 9 sub-spaces (categories), modify HGD structure to a classifier and plot its results as in Fig. 6 (supplementary, Sec. Regression and Classification). From the confusion matrix, we could see that HGD could accurately catch most of the gaze actions in practice.

5.4. Component Analysis

Significance of Head-pose Information: In Tab. 4, the first row represents our main proposed method, HGD. It demonstrates that when all components are included, HGD framework could achieve the best accuracy. From top to bottom, we in sequence get rid of MHoG + LDA, head-pose task and face model. As a result, we observe increasing gaze errors which demonstrate the importance of head-pose information in gaze estimation comparatively. Additionally, as listed in Tab. 5, incorporating head-pose information into gaze estimation could consistently gain improvements across different backbones. Our work focuses on proposing a novel framework for incorporating head-pose into gaze

estimation under two different scenarios regardless of backbones. Our solution is general to various backbone neural networks including Lenet, ResNet-34, ResNet-52, ResNet-101, ResNet-121, *etc*.

	Face Model	Head-Pose Task	mHoG + LDA	AEM	VEM
HGD	✓	✓	✓	**1.79**	**2.87**
	✓	✓	x	2.33	3.53
	✓	x	x	3.95	6.25
	x	x	x	6.88	8.48

Table 4: Comparison of HGD with various components on In-car Gaze dataset.

Backbone		ResNet10	ResNet18	**ResNet34**	ResNet56	ResNet101
w/ Face Model (head-pose information)	AEM	6.01	2.98	**1.79**	1.77	1.81
	VEM	8.07	4.67	**2.87**	2.83	2.85
w/o Face Model	AEM	4.23	3.92	**3.67**	3.69	3.65
	VEM	6.93	6.52	**5.28**	5.64	5.45

Table 5: Comparison of HGD framework with different backbone structures on In-car Gaze dataset. Despite that Resnet-56 and Resnet-101 may achieve slightly better accuracy than Resnet-34 in certain scenarios, we choose Resnet-34 as the main backbone due to its relatively much better computational efficiency.

Single Eye vs Double Eye: When a person is gazing at an object, both eyes have different gaze angles due to the distance between two pupils causing asymmetry. Gaze angles from both eyes should not be regarded as the same as assumed by many existing datasets [52, 38, 9]. This assumption would potentially risk the accuracy of gaze estimation. Under this insight, during collection, we purposely collect the specific gaze ground-truth labels for both eyes independently. To the best of our knowledge, our dataset is the only dataset that directly labels the difference between right eye's and left eye's gaze angles. Furthermore, we conduct extensive comparison experiments focusing on different means of merging both eyes during gaze estimation. These methods include: SEM, BEH, BEV and BEC. Note: SEM is the abbreviation for **S**ingle **E**ye **M**ethod where the algorithm only takes one eye at a time and outputs one gaze; BEH is the abbreviation for **B**oth **E**yes to be **H**orizontally stitched together and used as the input; BEV is the abbreviation for **B**oth **E**yes to be stitched together **V**ertically (supplementary, Sec. Merging Double Eyes). Since In-car Gaze is the only one that directly keeps different gaze labels for both eyes from the same face thus these comparison experiments could only be conducted on In-car Gaze, as in Tab. 3.

From the results, we conclude that BEC help the algorithm perform the best in both accuracy and computation efficiency. BEC could potentially find the correlation between both eyes in gaze estimation during training. When we conduct the comparison experiments on our full collected dataset, we note that BEC outperforms the SEM method by around 1 degree in accuracy. Furthermore, when we limit the training dataset to only 20,000 eye images (equivalent to 20,000 input units for SEM or 10,000 input units for BEC method), BEC method outperforms the SEM by almost 2 degrees in accuracy. We plot out the test gaze error graph

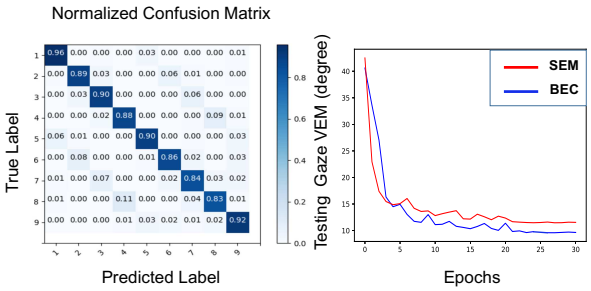

Figure 6: Left is a classification confusion matrix of HGD with explicit learning. The labeled number from 1 to 9 represents the 9 sub-spaces in front of the driver. Each sub-space represents an object, *e.g.* rear mirror (supplementary, Sec. Regression and Classification). Right is the test gaze error graph for both SEM and BEC methods within 30 epochs given limited 10000 units of data in our collected dataset. Under the same settings, test gaze error of BEC decreases faster than SEM. Best viewed in color.

for both SEM and BEC methods within 30 epochs given limited data and find out that, under the same settings, test gaze error of BEC would decrease faster than SEM, as in the right of Fig. 6. We believe this is due to the relatedness between right eye and left eye. This relatedness is easier for BEC to learn given both eyes from the same face especially when the training data is limited.

Different from [3] focusing on the difference, asymmetry of two eyes and trying to optimize gaze prediction through the better one between two streams, our methods try to learn the difference, asymmetry, through a single stream of fewer parameters. We believe the similarity between two eyes is substantial enough for the model to learn the difference.

Dataset	Method	AEM	VEM	Dataset	AEM	VEM	FLOPS(G)
In-car Gaze (full)	SEM	2.1	3.44	In-car Gaze (20,000 Eyes)	7.72	11.57	0.627
	BEH	2.04	3.33		7.31	11.08	0.624
	BEV	2.2	3.55		7.54	11.36	0.624
	BEC	**1.79**	**2.87**		**6.19**	**9.42**	**0.32**

Table 6: Comparison of using single or double eyes. BEC concatenates both eyes on the channel level. Thus the shape of input for both eyes would change from $2 \times W \times H \times C$ to $W \times H \times 2C$. For all the methods using both eyes as inputs, the algorithms would output the gaze angles of both eyes respectively and the final error is calculated by averaging both eyes' errors together.

6. Conclusion

In this work, we fully analyze the insufficiency of current methods and datasets on incorporating head-pose information into gaze estimation. We propose our frameworks that could better incorporate head-pose into gaze estimation in two scenarios. We further collect our own dataset to better evaluate our algorithms. Extensive evaluations demonstrate the advantage of our algorithms in free-head movement and our dataset in richer head-gaze distribution.

7. Acknowledgement

The writing of this paper was partially advised by Bo Wu.

References

[1] R. Bala, E. Bernal, A. Burry, P. Paul, and D. Chuang. Method and system for estimating gaze direction of vehicle drivers, Jan. 30 2018. US Patent 9,881,221.

[2] D. Beymer and M. Flickner. Eye gaze tracking using an active stereo head. In *Computer Vision and Pattern Recognition.*, volume 2, pages II–451, 2003.

[3] Y. Cheng, F. Lu, and X. Zhang. Appearance-based gaze estimation via evaluation-guided asymmetric regression. In *European Conference on Computer Vision*, pages 100–115, 2018.

[4] Z. R. Cherif, A. Nait-Ali, J. Motsch, and M. Krebs. An adaptive calibration of an infrared light device used for gaze tracking. In *IEEE Instrumentation and Measurement Technology Conference*, volume 2, pages 1029–1033, 2002.

[5] S. D'Mello, A. Olney, C. Williams, and P. Hays. Gaze tutor: A gaze-reactive intelligent tutoring system. *International Journal of Human-computer Studies*, 70(5):377–398, 2012.

[6] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2009.

[7] Z.-H. Feng, J. Kittler, M. Awais, P. Huber, and X.-J. Wu. Wing loss for robust facial landmark localisation with convolutional neural networks. In *Computer Vision and Pattern Recognition*, pages 2235–2245, 2018.

[8] T. Fischer, H. Jin Chang, and Y. Demiris. Rt-gene: Real-time eye gaze estimation in natural environments. In *European Conference on Computer Vision*, pages 334–352, 2018.

[9] K. A. Funes Mora, F. Monay, and J.-M. Odobez. Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras. In *Proceedings of the ACM Symposium on Eye Tracking Research and Applications*, pages 255–258, 2014.

[10] 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.

[11] D. Hansen and Q. Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:478–500, 03 2010.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[13] C. Hennessey, B. Noureddin, and P. Lawrence. A single camera eye-gaze tracking system with free head motion. In *Proceedings of Symposium on Eye Tracking Rresearch and Applications*, pages 87–94, 2006.

[14] Q. Huang, A. Veeraraghavan, and A. Sabharwal. Tabletgaze: dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets. *Machine Vision and Applications*, 28(5-6):445–461, 2017.

[15] A. Kar and P. Corcoran. A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms. *IEEE Access*, 5:16495–16519, 2017.

[16] A. Kar and P. Corcoran. Performance evaluation strategies for eye gaze estimation systems with quantitative metrics and visualizations. *Sensors*, 18(9):3151, 2018.

[17] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba. Eye tracking for everyone. In *Computer Vision and Pattern Recognition*, pages 2176–2184, 2016.

[18] M. Kumar, A. Paepcke, T. Winograd, and T. Winograd. Eyepoint: practical pointing and selection using gaze and keyboard. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 421–430, 2007.

[19] C.-C. Lai, S.-W. Shih, and Y.-P. Hung. Hybrid method for 3-d gaze tracking using glint and contour features. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(1):24–37, 2014.

[20] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[21] F. Lu, T. Okabe, Y. Sugano, and Y. Sato. A head pose-free approach for appearance-based gaze estimation. In *BMVC*, pages 1–11, 2011.

[22] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. Adaptive linear regression for appearance-based gaze estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(10):2033–2046, 2014.

[23] C. D. McMurrough, V. Metsis, J. Rich, and F. Makedon. An eye tracking dataset for point of gaze detection. In *Proceedings of the ACM Symposium on Eye Tracking Research and Applications*, pages 305–308, 2012.

[24] Y. K. Meena, H. Cecotti, K. Wong-Lin, A. Dutta, and G. Prasad. Toward optimization of gaze-controlled human–computer interaction: application to hindi virtual keyboard for stroke patients. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(4):911–922, 2018.

[25] A. Meyer, M. Böhme, T. Martinetz, and E. Barth. A single-camera remote eye tracker. In *International Tutorial and Research Workshop on Perception and Interactive Technologies for Speech-Based Systems*, pages 208–211, 2006.

[26] J. D. Morgante, R. Zolfaghari, and S. P. Johnson. A critical test of temporal and spatial accuracy of the tobii t60xl eye tracker. *Infancy*, 17(1):9–32, 2012.

[27] R. Naqvi, M. Arsalan, G. Batchuluun, H. Yoon, and K. Park. Deep learning-based gaze detection system for automobile drivers using a nir camera sensor. *Sensors*, 18(2):456, 2018.

[28] T. Ohno and N. Mukawa. A free-head, simple calibration, gaze tracking system that enables gaze-based interaction. In *Proceedings of the ACM Symposium on Eye Tracking Research and Applications*, pages 115–122, 2004.

[29] A. Palazzi, D. Abati, F. Solera, R. Cucchiara, et al. Predicting the driver's focus of attention: the dr (eye) ve project. *IEEE Transactions on Pattern Analysis and Machine intelligence*, 41(7):1720–1733, 2018.

[30] S. Park, A. Spurr, and O. Hilliges. Deep pictorial gaze estimation. In *European Conference on Computer Vision*, pages 721–738, 2018.

[31] S. Park, X. Zhang, A. Bulling, and O. Hilliges. Learning to find eye region landmarks for remote gaze estimation in unconstrained settings. In *Proceedings of the ACM Symposium on Eye Tracking Research and Applications*, page 21, 2018.

[32] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *Neural Information Processing Systems Autodiff Workshop*, 2017.

[33] H. F. Putra and K. Ogata. Development of eye-gaze interface system and its application to virtual reality controller. In *2018 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM)*, pages 208–213. IEEE, 2018.

[34] R. Ranjan, S. De Mello, and J. Kautz. Light-weight head pose invariant gaze tracking. In *Computer Vision and Pattern Recognition Workshop*, pages 2156–2164, 2018.

[35] R. L. Richmond, J. T. Haley, T. L. Schworer, M. W. Richmond, E. N. Richmond, O. S. Richmond, et al. Image changes based on viewer's gaze, Sept. 17 2019. US Patent App. 10/416,765.

[36] A. Schwarz, M. Haurilet, M. Martinez, and R. Stiefelhagen. Driveahead-a large-scale driver head pose dataset. In *Computer Vision and Pattern Recognition Workshop*, pages 1–10, 2017.

[37] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *Computer Vision and Pattern Recognition*, pages 2107–2116, 2017.

[38] B. A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar. Gaze locking: passive eye contact detection for human-object interaction. In *Proceedings of the ACM Symposium on User interface Software and Technology*, pages 271–280. ACM, 2013.

[39] Y. Sugano, M. Fritz, X. Andreas Bulling, et al. It's written all over your face: Full-face appearance-based gaze estimation. In *Computer Vision and Pattern Recognition Workshop*, pages 51–60, 2017.

[40] Y. Sugano, Y. Matsushita, and Y. Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *Computer Vision and Pattern Recognition*, pages 1821–1828, 2014.

[41] R. Valenti, N. Sebe, and T. Gevers. Combining head pose and eye location information for gaze estimation. *IEEE Transactions on Image Processing*, 21(2):802–815, 2011.

[42] A. Villanueva, V. Ponz, L. Sesma-Sanchez, M. Ariz, S. Porta, and R. Cabeza. Hybrid method based on topography for robust detection of iris center and eye corners. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 9(4):25, 2013.

[43] A. Vinciarelli, M. Pantic, and H. Bourlard. Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, 27(12):1743–1759, 2009.

[44] S. Vora, A. Rangesh, and M. M. Trivedi. Driver gaze zone estimation using convolutional neural networks: A general framework and ablative analysis. *IEEE Transactions on Intelligent Vehicles*, 3(3):254–265, 2018.

[45] K. Wang and Q. Ji. Real time eye gaze tracking with 3d deformable eye-face model. In *International Conference on Computer Vision*, pages 1003–1011, 2017.

[46] K. Wang, R. Zhao, H. Su, and Q. Ji. Generalizing eye tracking with bayesian adversarial learning. In *Computer Vision and Pattern Recognition*, pages 11907–11916, 2019.

[47] Y. Wang, G. Yuan, Z. Mi, J. Peng, X. Ding, Z. Liang, and X. Fu. Continuous driver's gaze zone estimation using rgb-d camera. *Sensors*, 19(6):1287, 2019.

[48] U. Weidenbacher, G. Layher, P.-M. Strauss, and H. Neumann. A comprehensive head pose and gaze database. 2007.

[49] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling. Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the ACM Symposium on Eye Tracking Research and Applications*, pages 131–138, 2016.

[50] Y. Yu, G. Liu, and J. Odobez. Improving few-shot user-specific gaze adaptation via gaze redirection synthesis. *CoRR*, abs/1904.10638, 2019.

[51] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

[52] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. Appearance-based gaze estimation in the wild. In *Computer Vision and Pattern Recognition*, pages 4511–4520, 2015.

[53] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):162–175, 2017.

[54] J. Zhu and J. Yang. Subpixel eye gaze tracking. In *International Conference on Automatic Face Gesture Recognition*, pages 131–136, 2002.

[55] W. Zhu and H. Deng. Monocular free-head 3d gaze tracking with deep learning and geometry constraints. In *International Conference on Computer Vision*, pages 3143–3152, 2017.

[56] Z. Zhu and Q. Ji. Novel eye gaze tracking techniques under natural head movement. *IEEE Transactions on Biomedical Engineering*, 54(12):2246–2260, 2007.