

Merging Multiple Datasets for Improved Appearance-Based Gaze Estimation

Liang Wu^[0000-0001-5214-7715] and Bertram E. Shi^[0000-0001-9167-7495]

Hong Kong University of Science and Technology, Hong Kong
liang.wu@connect.ust.hk
eebert@ust.hk

Abstract. Multiple datasets have been created for training and testing appearance-based gaze estimators. Intuitively, more data should lead to better performance. However, combining datasets to train a single estimator rarely improves gaze estimation performance. One reason may be differences in the experimental protocols used to obtain the gaze samples, resulting in differences in the distributions of head poses, gaze angles, illumination, etc. Another reason may be the inconsistency between methods used to define gaze angles (label mismatch). We propose two innovations to improve the performance of gaze estimation by leveraging multiple datasets, a change in the estimator architecture and the introduction of a gaze adaptation module. Most state-of-the-art estimators merge information extracted from images of the two eyes and the entire face either in parallel or combine information from the eyes first then with the face. Our proposed *Two-stage Transformer-based Gaze-feature Fusion* (TTGF) method uses transformers to merge information from each eye and the face separately and then merge across the two eyes. We argue that this improves head pose invariance since changes in head pose affect left and right eye images in different ways. Our proposed *Gaze Adaptation Module* (GAM) method handles annotation inconsistency by applying a Gaze Adaption Module for each dataset to correct gaze estimates from a single shared estimator. This enables us to combine information across datasets despite differences in labeling. Our experiments show that these innovations improve gaze estimation performance over the SOTA both individually and collectively (by 10% - 20%). Our code is available at <https://github.com/HKUST-NISL/GazeSetMerge>.

Keywords: gaze estimation · transformers · feature fusion · multi-dataset training.

1 Introduction

Estimation of human gaze plays important roles in many applications, such as human-computer interaction [2,3], virtual reality [1], attention analysis [4,5] and psychological studies [6].

Conventional methods, such as those based on pupil center corneal reflections (PCCR), use 3D eye models to compute the gaze direction [11]. These

require special measurement setups, such as active infrared illumination, to estimate model geometry. In contrast, appearance-based gaze estimators use input from commonly available RGB web cameras, which are more convenient and less expensive. Unfortunately, estimates from them are less accurate than those from PCCR-based systems. The current lowest reported within-person error of gaze estimation is 4.04° [42] on MPIIFaceGaze. In contrast, manufacturers of PCCR-based systems typically report accuracies of less than one degree.

However, the gap between the two continues to shrink, most recently due to the use of Convolutional Neural Networks (CNN) [7,8,9] and transformers. Many CNN architectures have been proposed for appearance-based gaze estimation. Zhang et al. employed a multi-modal model that used eye images and an estimated head pose vector as inputs to estimate gaze direction [7]. Later, they applied spatial weighting to feature maps from the face image to enhance information from eye regions [8]. Other studies used three separate pipelines to extract features from images of the head and the two eyes and then fused them to predict the gaze [9,12]. Merging information from the eyes and the face improves estimation accuracy.

Since appearance-based gaze estimators rely heavily on training data, many datasets have been proposed to train gaze estimators. Initial datasets were collected under fairly well-controlled and limited conditions (e.g., ranges of head poses and gaze angles). More recent datasets have been collected on conditions of increased diversity. The availability of more data can potentially increase the performance of appearance-based gaze estimators, but can also introduce new challenges. This paper seeks to address two of these challenges.

First, increases in the head pose range have spurred the development of new architectures that combine information from images of the two eye regions (which primarily indicate gaze direction in head-centric coordinates) and an image of the entire face (which primarily indicates head pose). Many SOTA (state-of-the-art) methods combine this information in parallel [9], or combine information from the eyes first followed by the face image [12].

To improve upon these approaches, we propose a Two-stage Transformer-based Gaze-feature Fusion (TTGF) architecture, which combines information from each eye image with the face image separately and then integrates information across the two eyes. This approach is motivated by the fact that the head-centric gaze directions of the two eyes differ and should thus each be merged with the face image. This may also compensate for situations where the reliability of information from the two eyes may differ, e.g., due to occlusion.

Second, although intuitively increasing the amount of data by combining datasets should improve performance, inconsistencies in annotation among datasets make it difficult to improve accuracy by simply combining multiple gaze datasets. To provide a normalized gaze annotation, a common scheme is to rotate the gaze vector from the gaze origin to the target point by a rotation matrix that depends upon the head pose [31]. Differences between the methods for head pose estimation and target point estimation lead to inconsistency among different datasets. Even when the subject’s head is constrained by a chin rest [29], head pose es-

timation error can still exist due to the placement of the subject’s head in the chin rest.

To address this, we propose the use of a Gaze Adaption Module (GAMs) for each dataset, which adjusts the gaze label from a shared estimator so it is consistent with the dataset of the source image. This enables multi-dataset training by simply adding GAM to the model’s gaze regression head.

Our experimental results demonstrate that these two innovations lead to state-of-the-art performance on multiple datasets, under training with both single datasets and mixed datasets.

2 Related Work

Gaze Estimation Methods Gaze estimation methods can typically be categorized as either model-based or appearance-based. Model-based methods usually construct the 3D model of the head and eyes. The gaze direction is calculated by utilizing geometric information [13,14,15,11]. Model-based methods usually require time-consuming personal calibration to fit the subject-specific parameters, such as cornea radius and kappa angles.

In contrast, appearance-based methods directly learn mapping functions from a large number of image-gaze sample pairs. Early approaches used conventional regression to perform the mapping [16,17,18]. More recently, CNNs have significantly improved the performance of appearance-based gaze estimation. Zhang et al. proposed the first CNN-based network to regress the gaze direction from a cropped eye image, and a head pose vector [7]. They later proposed to use the learnable spatial weights to enhance the information from the eye regions in the face image [8]. Krafka et al. proposed iTracker, a multi-region CNN model, which takes both the head and eye images as input. To further improve the accuracy, Chen et al. investigated the dilated convolution layers to efficiently increase the receptive field sizes of the features [9]. Researchers have now started to use transformer-based networks, which can further improve gaze estimation accuracy [19,20,21].

Transformers The Transformer architecture was first introduced by Vaswani et al. for natural language processing [38]. It consists of self-attention layers, layer normalization, and multi-layer perceptron layers. Compared with recurrent networks, the global computations and efficient memory of self-attention layers make transformers more suitable for long sequences.

The Vision Transformer (ViT) was proposed by Dosovitskiy et al. for image classification tasks [37]. ViT divides one image into non-overlapping patches. A transformer encoder is applied to the features extracted from the patches. Transformers have achieved state-of-the-art in large-scale image classification tasks, leading to their application to many other vision tasks [39,40,41].

Recently, a few researchers have explored the capability of transformers in gaze estimation. Cheng et al. proposed GazeTR-Hybrid where they used convolutional neural networks to extract the feature map of an input head image, then treated the features at different positions as a sequence of features input

to a transformer encoder [19]. Cai et al. proposed iTracker-MHSH [21]. Inspired by iTracker, it uses a transformer to integrate the features of the head and eye images.

Mixed Dataset Training There are two main advantages to mixed dataset training. First, it provides a single model applicable to multiple datasets. Second, model training may benefit from the increased amount of data. Mixed dataset training has been applied to many computer vision tasks, such as person reidentification [22,23], monocular depth estimation [24], semantic image segmentation [25,26], video quality assessment [27,30] and 3D object detection [28]. Addressing the challenges of mixed dataset training is task-specific. For example, to mix image segmentation datasets, category merging needs were conducted before training [25,26]. For video quality assessment [27], the challenge was to resolve inconsistent ranges of subjective quality scores across datasets.

To the best of our knowledge, we are the first to propose mixed dataset training for gaze estimation. There are two challenges that must be addressed. First, the distribution of gaze vectors and head poses varies between different gaze datasets. Second, there exists annotation inconsistency in gaze vectors from different gaze datasets.

3 Annotation Inconsistency

The gaze vector is defined as the vector starting from the gaze origin to the gaze target. Gaze dataset collection requires an experimental setup to capture three types of information in camera coordinates: 1) the position of the visual target P_t , 2) the position of gaze origin P_o , and 3) the head pose R [32]. However, different datasets utilize different methods to get these values, leading to different annotations.

Inconsistency in gaze target estimation Usually, the visual target is indicated by a moving dot on a screens. To determine the position of the dot target, the intrinsic parameters of the camera must be obtained beforehand. MPIIGaze uses a mirror-based calibration method [32] to estimate the 3D positions of each screen plane. Finally, the position of the moving dot is computed based on the screen size and resolution. In addition to a moving dot on the screen, EYEDIAP has an additional floating ball visual target. Its position is estimated first in an RGB-D sensor coordinate system and then transformed to the camera coordinate system. Imprecision in the RGB-D sensor, errors in the screen-to-camera calibration and RGB-D to-camera calibration will all contribute to the inconsistency of the gaze target position p_t .

Inconsistency in gaze origin and head pose estimation There are inconsistencies between datasets in the selection of gaze origin and the estimation of head pose. In early work, gaze was estimated eye images, where the eye center defined the gaze origin [7,35,36]. More recently, people estimate gaze from the whole head image, where the gaze origin is usually set at the center of the head [33,10,34]. To get the 3D head pose, MPIIGaze and ETH-XGaze detect landmarks from the 2D head image and fit a 3D morphable model of the head to the

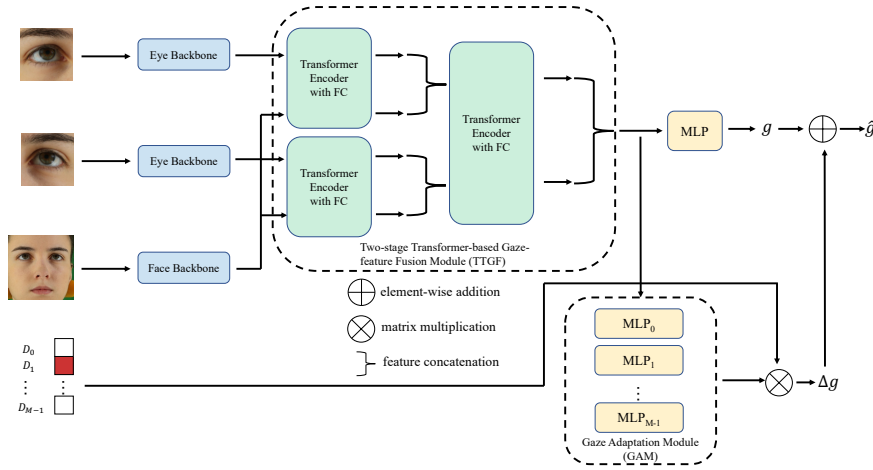


Fig. 1. The proposed framework contains two modules: 1) TTGF and 2) GAM. The TTGF applies two-stage feature fusion to the features of the head and eyes with transformers, and the GAM produces a gaze offset to adjust the predicted gaze for mixed datasets training.

detected landmarks. EYEDIAP directly uses the depth data from the RGB-D sensor to fit a 3D Morphable Model.

4 Method

Fig.1 shows our framework, which consists of an eye-head transformer-based feature fusion module for gaze estimation followed by a set of gaze adaptation modules. We described these in more detail below.

4.1 Feature Fusion with Transformers

A typical transformer encoder contains L transformer blocks, each containing multi-head self-attention (MHSA) layers, layer normalization (LN), and multi-layer perceptron layers (MLP). To process an input feature matrix $\mathbf{Z} \in \mathbb{R}^{n \times d}$, MHSA projects \mathbf{Z} into $\mathbf{Q} \in \mathbb{R}^{n \times d_k}$, keys $\mathbf{K} \in \mathbb{R}^{n \times d_k}$ and values $\mathbf{V} \in \mathbb{R}^{n \times d_v}$ where n is the number of tokens and d, d_k, d_v are the dimension of the feature, key/query and value.

The attention is computed through the following equation:

$$\text{Attention}(\mathbf{Q}; \mathbf{K}; \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\mathbf{V}\right). \quad (1)$$

Combined with LN and MLP, the overall equations for the transformer encoder with L transformer blocks are

$$z'_l = \text{MSA}(\text{LN}(z_{l-1})) + z_{l-1}, \quad l = 1 \dots L, \quad (2)$$

$$z_l = \text{MLP}(\text{LN}(z'_l)) + z'_l, \quad l = 1 \dots L, \quad (3)$$

$$y = \text{LN}(z_L). \quad (4)$$

Krafka et al. proposed iTracker [12] to estimate gaze by integrating the features of the head and eyes using several fully connected layers. To better fuse features, we propose the two-stage transformer-based gaze-feature fusion (TTGF) architecture shown in Fig. 1. This architecture applies three transformer encoders to fuse the features from the head and eye images in two fusion steps, 1) head-eye fusion and 2) left-right fusion. The idea of using two-step fusion is based on the intuition combining information of the head and one eye enable rough inference of the person’s gaze direction. The second step combines the two rough estimates into a single more precise estimate.

In our design, the architectures of all three fusion modules are identical. One TGF module accepts two gaze-related features and produces a fused feature. We describe the computation in a TGF formally with the following equation:

$$\text{TGF}(f^*, f^\dagger) = \text{CAT}(\text{FC}(\text{Trans}([f^*; f^\dagger])),) \quad (5)$$

where $\text{Trans}([f^*; f^\dagger])$ is the transformer used for fusing the head-eye features or eye-eye features, FC is a linear layer used to project the features to a specific size, and CAT concatenates the outputs of the transformer to generate fused features. In the head-eye fusion stage, each eye feature f^{le} or f^{re} is fused with the head feature f^h :

$$f^{lh} = \text{TGF}^{lh}(f^{le}, f^h) \quad (6)$$

$$f^{rh} = \text{TGF}^{rh}(f^{re}, f^h) \quad (7)$$

In the second stage, the two fused eye-head features are fed into a third TGF module to fuse features from left and right:

$$f^{lr} = \text{TGF}^{lr}(f^{lh}, f^{rh}) \quad (8)$$

Finally, the fused feature f^{lr} is fed to an MLP to get the predicted gaze g :

$$g = \text{MLP}(f^{lr}). \quad (9)$$

4.2 Gaze Adaptation Module

Suppose we have M gaze datasets, $D = \{D_0, D_2, \dots, D_{M-1}\}$. Typically, we need to train M models: one for each dataset to get good performance. A model trained on D_i typically performs poorly on D_j where $i \neq j$.

Instead, our approach trains only one model and $M - 1$ Gaze Adaptation Modules (GAMs). The GAM is a module consisting of a M MLPs, one for each

dataset $i \in \{0, \dots, M - 1\}$. Each MLP, $\text{MLP}_i(\cdot)$, accepts the extracted feature f^{lr} and produces a gaze offset assuming the sample comes from dataset i . D_0 is regarded as the anchor dataset, so its offset is always zero, i.e., $\text{MLP}_0(\cdot) = \mathbf{0}$ and does not need to be trained. For the others, the MLP has two layers with GELU nonlinearities. If the sample comes from dataset i , the corrected gaze vector is given by $\hat{g} = g + \Delta g$, where $\Delta g = \text{MLP}_i(f^{lr})$.

4.3 Architecture Details

The whole architecture contains three pipelines for the face and two eye images. All the backbones are ResNet18 networks, which are initialized from the model trained on ImageNet. The input face image size is $224 \times 224 \times 3$. We crop the eye patches according to the landmarks and use RoI align to resize the cropped patches to $128 \times 128 \times 3$. The estimated gaze contains the yaw and pitch representing the 3D gaze direction in the camera coordinate system. We chose L1 loss as the loss function for gaze estimation.

For TTGF, we set the number of heads of all MSAs as 8 and the hidden size of the MLP is 2048. We use 8 repeated blocks in each transformer encoder. After each transformer encoder, the features are projected with a linear layer whose output size is 128. For the MLPs for both gaze regression and the GAMs, the sizes of the hidden layers are identically set to 128.

5 Experiments

In this section, we introduce the experimental settings and the evaluation datasets we selected and evaluate our proposed TTGF and GAM in two types of experiments. We first compare our method with the state-of-the-art methods for gaze estimation performance. Then we perform ablation studies to determine the effects due to TTGF and GAM respectively and study the effect of multiple dataset training.

Dataset for evaluation For evaluating gaze estimation performance, we used three gaze datasets to evaluate the gaze estimation performance as shown in Table 1: MPIIFaceGaze [8], RT-GENE [10], and EYEDIAP [34]. MPIIFaceGaze dataset is based on MPIIGaze, but includes face and eye images. It contains 45K images collected from 15 subjects. We used leave-one-person-out cross-validation with this dataset. The RT-GENE dataset consists of 123K samples from 15 participants. We used three-fold cross-validation with this dataset. The raw data of the EYEDIAP dataset has 94 videos collected from 16 subjects. We used the sampling scheme from [33] to extract face images and four-fold cross-validation. For our experiments on multi-dataset training, we trained 15 models (one for each subject left out from MPIIFaceGaze), where each person was assigned to one of the folds in the other two datasets. Performance for each fold in the other two datasets was computed by averaging the performance of the models from the MPIIFaceGaze subjects assigned to that fold.

Table 1. Overview of the datasets used for evaluation and anchor dataset in our experiments. We show the number of subjects, the range of gaze, and the head pose in both horizontal and vertical directions in the camera coordinate systems.

Dataset	# Subjects	Gaze	Head Pose	# Data
MPIIFaceGaze [8]	15	$\pm 20^\circ, \pm 20^\circ$	$\pm 15^\circ, 30^\circ$	45K images
RT-GENE [10]	15	$\pm 40^\circ, -40^\circ$	$\pm 40^\circ, \pm 40^\circ$	123K images
EYEDIAP [34]	16	$\pm 25^\circ, 20^\circ$	$\pm 15^\circ, 30^\circ$	94 videos
ETH-XGaze [33]	110	$\pm 120^\circ, \pm 70^\circ$	$\pm 80^\circ, \pm 80^\circ$	1.1M images

Anchor dataset and pre-training ETH-XGaze [33] is a large-scale gaze dataset that consists of 1,083,492 image samples from 110 participants (47 female and 63 male). It has the largest range of head poses compared to the evaluation dataset and the gaze direction is evenly sampled both horizontally and vertically as shown in Table 1. The large variation and scale make it a suitable dataset as the anchor dataset D_0 and for pre-training. The whole dataset contains three parts: the training set, the within-dataset, and the person-specific evaluation set. The training set has 765K images of 80 subjects. We use this part as the anchor set and also for pre-training. The person-specific evaluation consists of 15 subjects but is not related to this task. The within-dataset which includes 15 subjects is used for validation of multiple datasets training and the pre-training model.

Experimental settings The optimizer applied for model training is AdamW with a linear scheduled warm-up strategy. The initial learning rate is set to 0.0001 for all the training and uses the exponential schedule to update it. For multiple-set training, in each iteration, we randomly sample the same number of samples from each set to form a batch fed to the model. The batch size is set to 64. The number of iterations in one training epoch is determined by the size of the dataset with the smallest number of samples. The number of epochs is 50 and gamma is 0.96. For single-set training for the TTGF-only model, the batch size is also set to 64. For ETH-XGaze, we train the model for 50 epochs with the exponential gamma setting to 0.95. For MPIIFaceGaze and RT-GENE, the total number of epochs is 30 epochs with the exponential gamma setting to 0.95. For EYEDIAP, the number of epochs is 50 and gamma is 0.096. Our experiments are all conducted on a single GeForce RTX 3090 GPU.

5.1 Comparison with state-of-the-art methods

In this part, we compare the gaze estimation performance of our proposed model with state-of-the-art methods. Our model is a single model trained on multiple datasets: one anchor dataset and three evaluation datasets, while the existing methods were tested with separated models for different evaluation datasets. We trained our TTGF-only model on ETH-XGaze and got a testing error of 3.58° and the proposed TTGF+GAM trained on multiple datasets achieved a slightly better error of 3.54° .

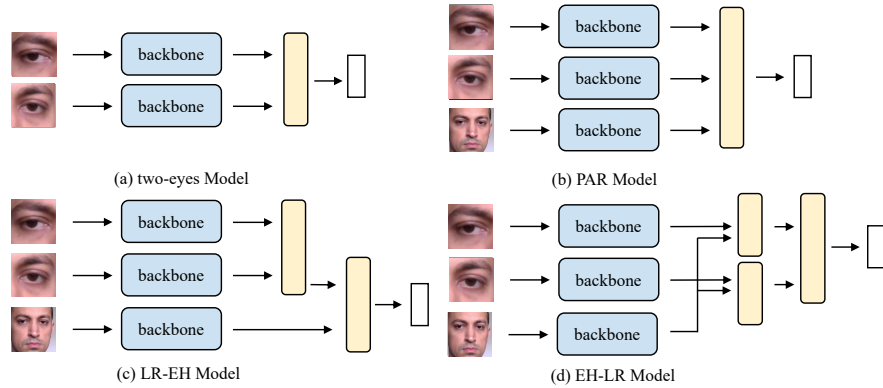


Fig. 2. Four types of feature fusion for gaze estimation models: (a) two-eyes model uses the cropped eye patches as inputs. (b) PAR indicates left eye, right eye, and head features are combined in parallel. (c) LR-EH indicates that left and right eye features are combined first then combined head features. (d) EH-LR indicates that single eye and head features are combined first followed by a combination across the left and right..

Table 2. Comparison with the state-of-the-art methods. The proposed method outperforms state-of-the-art results in estimation error.

Model	Transformer	Feature Fusion	MPIIFaceGaze	RT-GENE	EYEDIAP
FullFace [8]	NO	Face Only	4.93°	10.00°	6.53°
RT-GENE [10]	NO	Two Eyes	4.66°	8.00°	6.02°
DilatedNet [9]	NO	PAR	4.42°	8.38°	6.19°
iTracker [12]	NO	LR-EH	4.33°	7.12°	5.28°
iTracker-MHSA [21]	YES	LR-EH	4.05°	7.06°	5.17°
GazeTR-Hybrid [19]	YES	Face Only	4.18°	7.12°	5.33°
GazeCADSE [42]	YES	Face Only	4.04°	7.00°	5.25°
Proposed	YES	EH-LR	3.88°	6.46°	4.89°

Table 2 shows the angular errors of each method on the evaluation datasets: MPIIFaceGaze, RT-GENE, and EYEDIAP. As iTracker and iTracker-MHSA did not provide the performance on the evaluation datasets, we re-implemented them by replacing their backbones with ResNet18 for fair comparison. In the table, among existing models, FullFace [8], GazeTR [19], and GazeCADSE [42] only use the full face image as the input for gaze estimation. RT-GENE [10] feeds two cropped eyes to a VGG16 model. DilatedNet [9] fuses the features of the left eye, right eye, and head directly. iTracker [12], iTracker-MHSA [21] fuse the features of the left and right eyes first then with the head features. Our proposed method also uses both the face and the eye images as inputs but has different ways of feature fusion we fuse the features of each eye and head in the first stage and then fuse the left and right features in the second stage. In addition, GazeTR,

GazeCADSE, and our proposed methods utilize the transformers in the model. We show different types of gaze estimation models in Fig. 2.

As shown in Table 2, our proposed methods TTGF with GAM achieved the state-of-the-art performance of gaze estimation on all the selected evaluation datasets. Among the methods using the feature fusing, our eye-head first then left-right combination shows the best performance. Overall the transformer-based methods show advantages in the performance of gaze estimation compared with non-transformer methods. Among the transformer-based methods, our model uses both the face and eye images, we used RoI alignment to resize the eye region to 128×128 , which enables the model to extract features directly from the eye patches.

Table 3. Comparison of Computational Costs.

Model	Params	FLOPs
RT-GENE [10]	82.0M	30.81G
GazeTR-Hybrid [19]	11.4M	1.82G
GazeCADSE [42]	74.8M	12.78G
proposed method	65.3M	3.03G

By using GAM, our proposed model achieves better performance on multiple datasets using only a single main model. This results in a smaller number of parameters compared with other methods. Suppose the number of parameters of the feature extractor is N and that of each gaze regressor is K . For M datasets, without GAM we need to train M models for each dataset resulting in total MN parameters. On the contrary, by applying GAM to train on multiple datasets, we only need one single model with one feature extractor, one gaze regressor and $M - 1$ MLPs as the gaze offset for the anchor set is always $\mathbf{0}$. So the total number of parameters for our proposed model is $N + MK$. As K is much smaller than N , our method needs fewer parameters to achieve better performance.

Table 3 shows the number of parameters and the flops for each model. We can see that our proposed method has a fairly low computational cost which we believe is related to two reasons: 1) a relatively smaller model ResNet18 is applied as the backbone, and 2) a smaller size for the two eye patches as inputs.

5.2 Ablation Study

To study the individual contributions of the TTGF and GAM modules, we conducted ablation experiments by removing one of them from the entire framework.

Effect of TTGF To study the TTGF, we trained a TTGF-only model on each evaluation dataset and compared the results with itracker-MHSA. We compare with itracker-MHSA because it also uses a transformer encoder to combine eye and head features in a different order. The itracker-MHSA fuses features first

Table 4. Ablation study.

Model	Multiple Sets	MPIIFaceGaze	RT-GENE	EYEDIAP
itracker-MHSA [21]	NO	4.05 °	7.06°	5.17°
TTGF-Only	NO	3.98°	6.89°	5.11°
TTGF-Only	YES	4.12 °	7.14 °	5.20°
proposed method (TTGF+GAM)	YES	3.88°	6.46°	4.89°

from the left and right eyes and then with the head feature. TTGF fuses features from each eye with head features and then across the two eyes. As we mentioned before, we re-implemented itracker-MHSA with the same backbone as our model for a fair comparison. As we mentioned before, we re-implemented itracker-MHSA with the same backbone as our model for a fair comparison.

Table 4 shows the angular errors of each method on the evaluation datasets. The TTGF-only model outperforms the itracker-MHSA on all evaluation datasets.

Effect of GAM We compared our proposed model with GAM with the TTGF-only model trained on multiple datasets. Table 4 shows that with GAM the accuracy of the TTGF-only model without multiple sets of training is improved on all three datasets from 0.1° to 0.43° respectively.

To confirm the performance gain in multiple dataset training is due to the use of GAM, we trained the TTGF-only model with the combination of the ETH-XGaze and the evaluation datasets. The TTGF-only model trained on mixed datasets performed even worse than the TTGF-only model trained on each single evaluation set. This supports our claim that GAM can address the inconsistency in annotation across different datasets.

6 Conclusion

We proposed a Two-stage Transformer-based Gaze-future Fusion (TTGF) and the use of Gaze Adaption Modules (GAMs) for improving gaze estimation accuracy. The TTGF uses two-stage fusion for the features of the head and eye images through three transformer encoders. The proposed GAM generates gaze corrections to gaze estimates for one dataset (chosen here to be ETH-Gaze) to create estimates for images from other datasets. Our experiments show that our method surpasses the state-of-the-art by a significant margin. Ablation studies show that both innovations result in improvements when applied in isolation and that improvements compound when they are applied together. However, our proposed model still has some limitations. For example, the proposed TTGF needs cropped eye patches as input. The GAM does not address all issues arising from annotation inconsistency among gaze datasets.

References

1. Patney, A., Salvi, M., Kim, J., Kaplanyan, A., Wyman, C., Benty, N., Luebke, D. & Lefohn, A. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions On Graphics (TOG)*. **35**, 1-12 (2016)
2. Chen, Z. & Shi, B. Using variable dwell time to accelerate gaze-based web browsing with two-step selection. *International Journal Of Human-Computer Interaction*. **35**, 240-255 (2019)
3. Pi, J. & Shi, B. Probabilistic adjustment of dwell time for eye typing. *2017 10th International Conference On Human System Interactions (HSI)*. pp. 251-257 (2017)
4. Recasens, A., Khosla, A., Vondrick, C. & Torralba, A. Where are they looking?. *Advances In Neural Information Processing Systems*. **28** (2015)
5. Chong, E., Wang, Y., Ruiz, N. & Rehg, J. Detecting attended visual targets in video. *Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition*. pp. 5396-5406 (2020)
6. Gehrer, N., Schöenberg, M., Duchowski, A. & Krejtz, K. Implementing innovative gaze analytic mods in clinical psychology: A study on eye movements in antisocial violent offenders. *Proceedings Of The 2018 ACM Symposium On Eye Tracking Research & Applications*. pp. 1-9 (2018)
7. Zhang, X., Sugano, Y., Fritz, M. & Bulling, A. Appearance-based gaze estimation in the wild. *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition*. pp. 4511-4520 (2015)
8. Zhang, X., Sugano, Y., Fritz, M. & Bulling, A. It's written all over your face: Full-face appearance-based gaze estimation. *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition Workshops*. pp. 51-60 (2017)
9. Chen, Z. & Shi, B. Appearance-based gaze estimation using dilated-convolutions. *Asian Conference On Computer Vision*. pp. 309-324 (2018)
10. Fischer, T., Chang, H. & Demiris, Y. Rt-gene: Real-time eye gaze estimation in natural environments. *Proceedings Of The European Conference On Computer Vision (ECCV)*. pp. 334-352 (2018)
11. Guestrin, E. & Eizenman, M. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions On Biomedical Engineering*. **53**, 1124-1133 (2006)
12. Krafska, K., Khosla, A., Kellnhofer, P., Kannan, H., Bhandarkar, S., Matusik, W. & Torralba, A. Eye tracking for everyone. *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition*. pp. 2176-2184 (2016)
13. Chen, J. & Ji, Q. 3D gaze estimation with a single camera without IR illumination. *2008 19th International Conference On Pattern Recognition*. pp. 1-4 (2008)
14. Valenti, R., Sebe, N. & Gevers, T. Combining head pose and eye location information for gaze estimation. *IEEE Transactions On Image Processing*. **21**, 802-815 (2011)
15. Wood, E. & Bulling, A. Eytat: Model-based gaze estimation on unmodified tablet computers. *Proceedings Of The Symposium On Eye Tracking Research And Applications*. pp. 207-210 (2014)
16. Tan, K., Kriegman, D. & Ahuja, N. Appearance-based eye gaze estimation. *Sixth IEEE Workshop On Applications Of Computer Vision, 2002. (WACV 2002). Proceedings..* pp. 191-195 (2002)
17. Lu, F., Sugano, Y., Okabe, T. & Sato, Y. Adaptive linear regression for appearance-based gaze estimation. *IEEE Transactions On Pattern Analysis And Machine Intelligence*. **36**, 2033-2046 (2014)

18. Williams, O., Blake, A. & Cipolla, R. Sparse and Semi-supervised Visual Mapping with the Sundefined 3GP. *2006 IEEE Computer Society Conference On Computer Vision And Pattern Recognition (CVPR)*. **1** pp. 230-237 (2006)
19. Cheng, Y. & Lu, F. Gaze estimation using transformer. *2022 26th International Conference On Pattern Recognition (ICPR)*. pp. 3341-3347 (2022)
20. Tu, D., Min, X., Duan, H., Guo, G., Zhai, G. & Shen, W. End-to-End Human-Gaze-Target Detection with Transformers. *ArXiv Preprint arXiv:2203.10433*. (2022)
21. Cai, X., Chen, B., Zeng, J., Zhang, J., Sun, Y., Wang, X., Ji, Z., Liu, X., Chen, X. & Shan, S. Gaze Estimation with an Ensemble of Four Architectures. *ArXiv Preprint arXiv:2107.01980*. (2021)
22. Lv, J., Chen, W., Li, Q. & Yang, C. Unsupervised cross-dataset person re-identification by transfer learning of spatial-temporal patterns. *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition*. pp. 7948-7956 (2018)
23. Li, Y., Lin, C., Lin, Y. & Wang, Y. Cross-dataset person re-identification via unsupervised pose disentanglement and adaptation. *Proceedings Of The IEEE/CVF International Conference On Computer Vision*. pp. 7919-7929 (2019)
24. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K. & Koltun, V. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions On Pattern Analysis And Machine Intelligence*. (2020)
25. He, H., Zhang, J., Zhang, Q. & Tao, D. Grapy-ML: Graph Pyramid Mutual Learning for Cross-Dataset Human Parsing. *The Thirty-Fourth AAAI Conference On Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications Of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium On Educational Advances In Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. pp. 10949-10956 (2020), <https://ojs.aaai.org/index.php/AAAI/article/view/6728>
26. Lambert, J., Liu, Z., Sener, O., Hays, J. & Koltun, V. MSeg: A composite dataset for multi-domain semantic segmentation. *Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition*. pp. 2879-2888 (2020)
27. Li, D., Jiang, T. & Jiang, M. Unified quality assessment of in-the-wild videos with mixed datasets training. *International Journal Of Computer Vision*. **129**, 1238-1257 (2021)
28. Zhang, W., Li, W. & Xu, D. SRDAN: Scale-aware and range-aware domain adaptation network for cross-dataset 3D object detection. *Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition*. pp. 6769-6779 (2021)
29. Smith, B., Yin, Q., Feiner, S. & Nayar, S. Gaze locking: passive eye contact detection for human-object interaction. *Proceedings Of The 26th Annual ACM Symposium On User Interface Software And Technology*. pp. 271-280 (2013)
30. Korhonen, J. Two-level approach for no-reference consumer video quality assessment. *IEEE Transactions On Image Processing*. **28**, 5923-5938 (2019)
31. Zhang, X., Sugano, Y. & Bulling, A. Revisiting data normalization for appearance-based gaze estimation. *Proceedings Of The 2018 ACM Symposium On Eye Tracking Research & Applications*. pp. 1-9 (2018)
32. Rodrigues, R., Barreto, J. & Nunes, U. Camera pose estimation using images of planar mirror reflections. *European Conference On Computer Vision*. pp. 382-395 (2010)
33. Zhang, X., Park, S., Beeler, T., Bradley, D., Tang, S. & Hilliges, O. Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. *European Conference On Computer Vision*. pp. 365-381 (2020)

34. Funes Mora, K., Monay, F. & Odobez, J. Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras. *Proceedings Of The Symposium On Eye Tracking Research And Applications*. pp. 255-258 (2014)
35. Park, S., Spurr, A. & Hilliges, O. Deep pictorial gaze estimation. *Proceedings Of The European Conference On Computer Vision (ECCV)*. pp. 721-738 (2018)
36. Liu, G., Yu, Y., Mora, K. & Odobez, J. A differential approach for gaze estimation. *IEEE Transactions On Pattern Analysis And Machine Intelligence*. **43**, 1092-1099 (2019)
37. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. & Others An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv Preprint arXiv:2010.11929*. (2020)
38. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, Ł. & Polosukhin, I. Attention is all you need. *Advances In Neural Information Processing Systems*. **30** (2017)
39. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. & Luo, P. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances In Neural Information Processing Systems*. **34** pp. 12077-12090 (2021)
40. Kim, B., Lee, J., Kang, J., Kim, E. & Kim, H. Hotr: End-to-end human-object interaction detection with transformers. *Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition*. pp. 74-83 (2021)
41. Zhang, W., Qiu, F., Wang, S., Zeng, H., Zhang, Z., An, R., Ma, B. & Ding, Y. Transformer-based Multimodal Information Fusion for Facial Expression Analysis. *Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition*. pp. 2428-2437 (2022)
42. O Oh, J., Chang, H. & Choi, S. Self-Attention With Convolution and Deconvolution for Efficient Eye Gaze Estimation From a Full Face Image. *Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition*. pp. 4992-5000 (2022)