FUSING COMPLEMENTARY MULTI-VIEW FEATURE FOR SCREEN-BASED EYE TRACKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Current multi-view gaze estimation remains limited by existing datasets, insufficient exploitation of complementary cross-view information, and evaluation focused primarily on average gaze error. We address these limitations through a more systematic study of multi-view gaze estimation. First, we introduce PrismGaze, a new dataset with over three million images, capturing continuous head-pose variation for the same gaze targets. Second, we propose PrismFusion, a multi-view feature fusion framework based on region partitioning, which masks complementary image regions across views during training to encourage effective cross-view information integration. Third, we develop a broader evaluation framework that examines the effects of camera number and placement, target location, and viewing depth. Our experiments show that the primary benefit of multi-view gaze estimation comes from compensating for poorly observed views with cameras providing more favorable viewpoints. PrismFusion remains robust to changes in viewing depth. Together, our dataset, method, and evaluation provide a more comprehensive foundation for studying multi-view gaze estimation in realistic settings.

1 INTRODUCTION

Gaze estimation or eye tracking is important to human–computer interaction and behavioral analysis, giving it great potential for building intelligent environments. Dedicated eye trackers provide accurate measurements, but their specialized sensors and per-user calibration limit unobtrusive deployment at scale (Bao & Lu, 2024). Instead, an increasing number of work now estimate gaze directly from RGB images. They have enabled gaze estimation during laptop use (Zhang et al., 2015), on mobile devices (Krafka et al., 2016), and from ordinary video cameras (Park et al., 2020). However, most existing methods rely on a single camera, making gaze estimation sensitive to camera placement, face or eye occlusion, and distance change between the user and the camera.

Multi-view eye tracking address several limitations of single-view methods by incorporating complementary information from extra views, leading to improved performance (Lian et al., 2019; Cheng & Lu, 2023). Beyond that, multi-view information has also been exploited to enhance generalization to unseen viewpoints (Hisadome et al., 2024; Liu & Lu, 2024). With calibrated cameras, multi-view geometry may reduce the need for session-specific calibration. Depth can be recovered from the disparity from views, and optical and visual axis can be estimated with only one calibration point (Shih & Liu, 2004).

Despite this progress, current multi-view eye tracking still face several challenges in data, methodology, and evaluation. From the **data** perspective, existing studies rely heavily on a small number of datasets: Eth-XGaze and MPII-NV (Zhang et al., 2020; Qin et al., 2022). ETH-XGaze provides diverse camera viewpoints and broad subject coverage, but its public available data mainly consist of normalized face images with users typically sitting on fixed positions, which limits the explicit use of depth data. Furthermore, each gaze target contains only a single image, providing limited variation in head pose and eye movement: a target on the left side of the screen is often associated with both a leftward head pose and a leftward gaze. In real-world use, however, users may look at the same target with very different head poses, such as turning the head to the right while looking left. Missing such variations may reduce robustness in unconstrained settings. From the **methodological** perspective, current fusion method may not fully exploit complementary information across

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

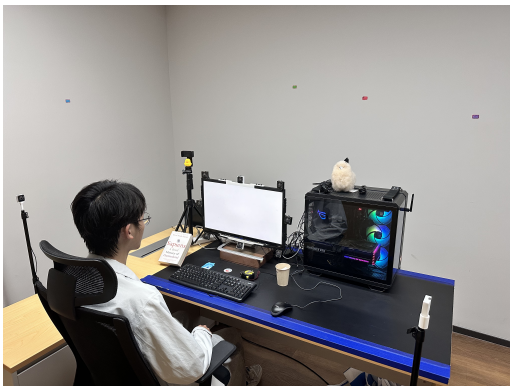
views. Although recent studies commonly use Transformer-like architectures for multi-view fusion, attention alone does not guarantee all necessary information from different information are kept and fully utilized. In our preliminary experiments, straightforward Transformer-based fusion showed similar performance on gaze estimation error compared to averaging independent estimation from different views, suggesting that the additional views are not effectively used. From the **evaluation** perspective, current evaluation often focus primarily the overall average of gaze estimation error, overlooking some factors that can substantially effect accuracy in practice, such as viewing distance, camera placement, and target position. More comprehensive evaluation is therefore needed to characterize when and where multi-view gaze estimation is effective.

In this work, we address this challenges through a more systematic formulation of multi-view gaze estimation. First, we introduce PrismGaze, a new multi-view gaze dataset containing 24 participants, 288 gaze targets, eight cameras (plus extra two side cameras), and 45 frames per target. Unlike existing datasets with limited head–eye variation for each target, PrismGaze captures continuous variation in head pose across frames. Second, we propose a multi-view feature fusion framework PrismFusion based on region partitioning. During training, each camera retains only a subset of the original image while the remaining regions are masked, encouraging Transformer to integrate complementary information across views. Finally, we introduce a broader evaluation framework. We study the effects of camera number and placement, target location, and viewing depth. Our results show that the main benefit of PrismFusion comes from compensating for poorly observed views with cameras that provide more favorable viewpoints. PrismFusion also remains robust to changes in viewing depth, while maintaining strong calibration-free performance on seen views.

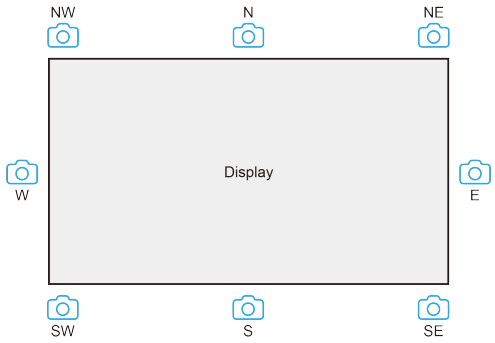
2 Related Works

Single-view eye tracking can be generally categorized into model-based or appearance-based method (Zhang et al., 2015; Hansen & Ji, 2009). Model-based method constructs geometric eye model. PCCR (Pupil Center Corneal Reflection) is typically used, which creates corneal glints and estimate gaze by mapping the vector between pupil center and glint to the corresponding gaze coordinate on the screen (Merchant et al., 1974; Morimoto & Mimica, 2005; Kim et al., 2026). Model-based gaze estimation explicitly reconstructs ocular geometry (e.g., pupil center, iris ellipse) from eye images to infer gaze direction (Chen & Ji, 2008; Kaur et al., 2022). The appearance-based method adopts an end-to-end manner that takes an RGB image as an input and outputs either the 3D gaze direction or the exact gaze position on the screen. Many annotated datasets have been collected, whose ground truth can be either the gaze location on screen or the 3d gaze vector (Zhang et al., 2015; Funes Mora et al., 2014; Evdokimov et al., 2024).

Multi-view camera settings have also been explored for eye tracking. Lian et al. (2019) combines features from three fixed cameras to estimate gaze direction and screen position. Its fixed input structure, however, does not address changes in the number or identity of the available cameras. DVGaze introduces structured cross-view correspondence for feature fusion, but is limited to a dual-camera setting (Cheng & Lu, 2023). Hisadome et al. (2024) use the relative geometry between cameras to guide feature transformation in latent space, enabling gaze estimation to generalize from seen to unseen camera views. UVAGaze achieves unsupervised adaptation to a new view by selecting more reliable prediction from the two views to supervise the other (Liu & Lu, 2024). Many methods have been proposed to fuse the multi-view information. Bao & Lu (2024) uses eye region mask and feature switch between different views to prompt better gaze representation learning. Similarly, Gideon et al. (2022) and Sun et al. (2021) leveraged multi-view consistency to learn disentangled gaze representations through latent feature swapping. Available datasets offer multi-view images with labels. EVE records synchronized videos from four user-facing cameras, with gaze label provided by an eye tracker (Park et al., 2020). ETH-XGaze captures broad head-pose and gaze variation with an 18-camera system (Zhang et al., 2020). However it provides only cropped face images, which limits the depth estimation. MPII-NV (Qin et al., 2022) reconstructs the 3D facial geometry from the single view dataset MPIIFaceGaze (Zhang et al., 2017), yet its views are consistent with the ones in ETH-XGaze.



(a) Experimental environment.



(b) Camera and screen setup.

Figure 1: Experimental setup for PrismGaze data collection. (a) Participants sit approximately 60 cm from the display while ten synchronized RGB cameras and a Tobii Eye Tracker 5 record each fixation trial. (b) Eight cameras are mounted around the display at the NW, N, NE, W, E, SW, S, and SE positions; two additional side-view cameras are placed to the left and right of the participant.

3 Constructing PrismGaze

3.1 Setup

The setup for data collection is shown in Fig. 1. Eight RGB web cameras are rigidly mounted at fixed locations along the display boundary, providing synchronized facial observations from different angles. Besides that, we also mounted two side-view RGB web cameras on the left and right sides of the participant to capture more extreme viewing angles. All cameras are synchronized and operate at 3840×2160 resolution at 30 FPS with identical settings.

Participants are seated approximately 60cm from a 1920×1080 display (54cm×32.5cm) and are instructed to fixate on predefined targets located either on or outside the screen. For on-screen targets, we define 288 locations arranged in a regular 24×12 grid covering the entire screen area. Each target corresponds to one fixation trial. To collect negative samples, we additional define 12 off-screen targets using physical objects placed on the desk or wall around the participant.

Each fixation trial lasts approximately 1.5 seconds. After fixating on the target, participants are instructed to gently and randomly rotate their head while maintaining their gaze on the target, allowing us to capture the same target under varying gaze and head poses. Each trial is initiated by the participant using keyboard input. During each trial, we record 45 synchronized frames from all camera streams, along with gaze estimates from a Tobii Eye Tracker 5.

The dataset contains 24 participants (13 F, 11M, age 23.15±3.35). All participants are required to remove glasses during data collection to reduce occlusion. Each participant completes 288 screen-based trials and 12 out-of-screen trials, resulting in 7,200 trials in total. Since each trial contains 45 synchronized frames recorded from ten cameras, the dataset contains approximately 3.24 million raw images.

3.2 Annotation

For eight camera on the boundary, the ground truth of on-screen target is defined as the 2D screen coordinate of the target, and that of off-screen target is defined as a out-screen label.

For the side-view camera, we define the ground-truth as certain area in image. For on-screen targets, we project the predefined 24×12 screen grid into each side-view camera image using a fixed projective transformation. The visible screen region is masked in the input image to prevent the model from directly inferring the target from the displayed content. For off-screen targets, the center of each predefined physical object is used as the corresponding ground-truth location (Details in Sec A.1).

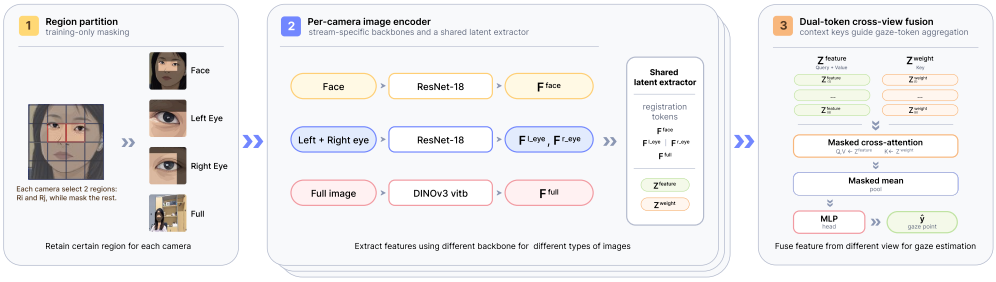


Figure 2: Overview of PrismFusion. Region Partition distributes cells from a 4×4 face grid across at least two retained cameras and applies the corresponding masks to the full-image, face, and eye crops. A per-view DINO–ResNet encoder produces feature and weight tokens. Cross-view attention uses Z^{feature} as queries and values and Z^{weight} as keys, followed by mean pooling and an MLP for gaze prediction.

3.3 Data Preprocess

To better support gaze estimation, we crop the original image from on-screen camera for each frame. Four different image are obtained using MediaPipe[1] and compressed into 256×256 image: full image, face crop, left eye, right eye. For images in which MediaPipe failed to detect a face, we performed additional detection using 3DDFA (Bulat & Tzimiropoulos, 2017) and InsightFace[2]. Images for which all three models failed were manually annotated.

We split the dataset into training, validation, and test sets at a 70%, 20%, 10% ratio using a subject-wise split, ensuring that participants in the three sets were mutually exclusive.

4 Method

4.1 Problem Formulation

Given synchronized observations from K cameras, our goal is to estimate the two-dimensional screen-space gaze target $\mathbf{y} \in \mathbb{R}^2$. For camera k, the input comprises four aligned image streams:

$$\mathcal{I}_k = \left\{ I_k^{\text{full}}, I_k^{\text{face}}, I_k^{\text{leye}}, I_k^{\text{reye}} \right\},$$

corresponding to the full image, face crop, left-eye crop, and right-eye crop. PrismFusion encodes global and local appearance cues within each view and fuses the camera-wise representations to predict $\mathbf{y}$.

4.2 Region Partition

With complete inputs from every camera, the model can rely on one favorable view or redundant cues across views. Region Partition distributes visible regions across cameras during training and promotes cross-view integration.

For each sample, camera dropout retains at least two cameras. The face crop is divided into a 4×4 grid whose cells are assigned randomly and approximately evenly among the retained cameras. Pixels outside a camera's assigned cells are masked, and the same ownership map is projected to the full-image and eye crops using their crop coordinates. Region Partition is applied only during training; inference uses complete images from all available cameras.

[1] https://developers.google.com/edge/mediapipe/solutions/vision/face_landmarker

[2] https://github.com/deepinsight/insightface

4.3 Per-View Feature Extraction

For each camera, the full image is encoded by a DINOv3 ViT-S backbone, the face crop is encoded by a ResNet, and the left- and right-eye crops are processed by another shared ResNet. We project the multi-scale ResNet features into the DINO feature space and convert them into tokens compatible with the DINO patch representation, yielding four feature-token sequences:

$$\mathbf{F}_k^{\text{full}}, \quad \mathbf{F}_k^{\text{face}}, \quad \mathbf{F}_k^{\text{leye}}, \quad \mathbf{F}_k^{\text{reye}}.$$

The four sequences are concatenated with two learnable query tokens, Z^{feature} and Z^{weight}, and four learnable register tokens. A self-attention encoder mixes the image and latent tokens within each view and outputs two camera-level representations:

$$\mathbf{z}_k^{\text{feature}}, \quad \mathbf{z}_k^{\text{weight}} \in \mathbb{R}^D,$$

These tokens provide the camera-level representations used for cross-view fusion.

4.4 Cross-View Fusion

For the retained camera set $\mathcal{V}$, the two representations form separate cross-view sequences:

$$\mathbf{Z}^{\text{feature}} = \left[\mathbf{z}_k^{\text{feature}}\right]_{k\in\mathcal{V}}, \qquad \mathbf{Z}^{\text{weight}} = \left[\mathbf{z}_k^{\text{weight}}\right]_{k\in\mathcal{V}}.$$

We employ a stack of cross-view attention blocks in which $\mathbf{Z}^{\text{feature}}$ provides the queries and values, while $\mathbf{Z}^{\text{weight}}$ provides the keys. Let $\mathbf{H}^{(0)} = \mathbf{Z}^{\text{feature}}$; each fusion block updates the camera-wise representation as

$$\mathbf{H}^{(\ell+1)} = \text{FFN}\Big(\mathbf{H}^{(\ell)} + \text{Attn}\Big(\mathbf{H}^{(\ell)}, \mathbf{Z}^{\text{weight}}, \mathbf{H}^{(\ell)}\Big)\Big),$$

The camera-wise features are mean-pooled over retained views and mapped by an MLP to the gaze prediction $\hat{\mathbf{y}}$. Training minimizes the Euclidean distance between the predicted and ground-truth gaze locations over $\mathcal{D}$:

$$\mathcal{L}_{\text{gaze}} = \frac{1}{|\mathcal{D}|} \sum_{(\mathcal{I},\mathbf{y})\in\mathcal{D}} \|\hat{\mathbf{y}} - \mathbf{y}\|_2 .$$

5 Evaluation

5.1 Experimental Setup

We evaluate camera-count effects on both PrismGaze and ETH-XGaze, camera placement and target location on PrismGaze, and viewing distance on an additional real-world dataset. PrismGaze uses a 70%/20%/10% subject-wise split with no participant overlap. For ETH-XGaze, 10 participants from the official training set are randomly selected for validation, the remaining 70 are used for training, and the official test set is used for evaluation. The checkpoint with the lowest validation error is evaluated on the corresponding test set. Unless otherwise stated, we report mean gaze-prediction error over all evaluation samples.

5.2 Performance on PrismGaze and ETH-XGaze

For methods with public code, we adapt the released implementations to match our input and output formats. PrismFusion has the lowest mean error in both the eight-view and E/W two-view comparisons in Table 1.

Table 2 reports published ETH-XGaze results together with PrismFusion. The all-view setting uses 18 cameras, the 10-view setting uses the inner ring, and the two-view setting follows the nine camera pairs used by UVAGaze (Liu & Lu, 2024): $(0, 9)$, $(1, 10)$, $(2, 11)$, $(3, 12)$, $(4, 13)$, $(5, 14)$, $(6, 15)$, $(7, 16)$, and $(8, 17)$. The 18-view setting has higher error than the 10-view setting under both metrics, showing that using all available views does not necessarily improve performance.

Method	Implementation	# Cams	Error (cm)
PrismFusion	Ours	8	**0.94**
Lian et al. (2019)	Reimplemented	8	1.46
DVGaze (Cheng & Lu, 2023) (8-view)	Extended from official code	8	1.33
GazeTR Cheng & Lu (2022) (8-view)	Extended from official code	8	2.49
PrismFusion (2-view)	Ours	2	**1.72**
Rot-MVGaze (Hisadome et al., 2024)	Official code	2	5.21
UVAGaze (Liu & Lu, 2024)	Official code	2	7.27
DVGaze (Cheng & Lu, 2023)	Official code	2	2.42

Table 1: Comparison on PrismGaze. We report mean two-dimensional screen-space error in centimeters (lower is better) for the eight-view setting and the E/W two-view setting. Bold values mark the best result within each setting.

Method	# Cameras	Error Within (°)	Error Person Specific (°)
PrismFusion (10 views)	10	2.89	3.25
PrismFusion (2 views)	2	3.14	3.00
PrismFusion (all views)	18	3.49	3.80
Rot-MVGaze (Hisadome et al., 2024)	2	3.5[†]	
UVAGaze (Liu & Lu, 2024)	2	3.27	
DVGaze (Cheng & Lu, 2023)	2	2.46[‡]	

Table 2: Angular gaze error on ETH-XGaze in degrees (lower is better). PrismFusion is evaluated with the 10 inner-ring cameras, nine two-view camera pairs, and all 18 cameras. Results for prior methods are quoted from their respective papers; as their evaluation protocols differ from ours, their values are placed between the two error columns rather than assigned to either. [†]Evaluated with 4-fold subject cross-validation. [‡]Potential data leakage identified through our inspection of the publicly available source code.

5.3 Ablation Studies

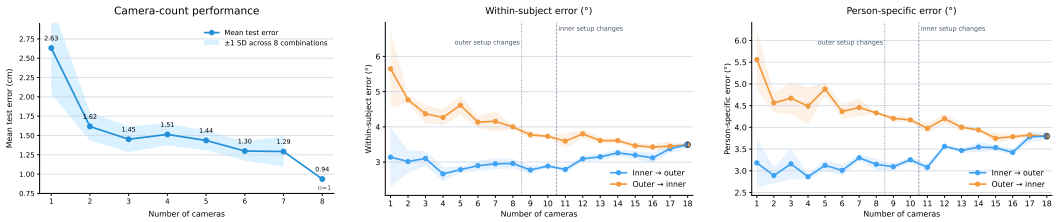
Removing Region Partition while keeping the backbone configuration fixed increases error from 0.94 cm to 1.63 cm (Table 3). Changing the ResNet assignments produces errors from 1.38 cm to 1.82 cm, while using DINO backbones for all inputs gives substantially higher errors. Region Partition and the mixed backbone design form the final configuration.

Index	Region Partition	DINO Finetune	Full	Face	Left Eye	Right Eye	Error (cm)
0	Yes	No	DINO	ResNet^0	ResNet^1	ResNet^1	**0.94**
1	No	No	DINO	ResNet^0	ResNet^1	ResNet^1	1.63
2	Yes	No	DINO	ResNet^0	ResNet^1	ResNet^2	1.47
3	Yes	No	ResNet^0	ResNet^1	ResNet^2	ResNet^2	1.38
4	Yes	No	ResNet^0	ResNet^1	ResNet^2	ResNet^3	1.82
5	Yes	No	DINO^0	DINO^1	DINO^2	DINO^2	6.05
6	Yes	No	DINO^0	DINO^1	DINO^2	DINO^3	6.35
7	Yes	Yes	DINO^0	DINO^1	DINO^2	DINO^2	3.58
8	Yes	Yes	DINO^0	DINO^1	DINO^2	DINO^3	6.87

Table 3: Ablation of Region Partition, DINO fine-tuning, and backbone assignments on PrismGaze. We report mean screen-space error in centimeters (lower is better); the bold row is the final PrismFusion configuration. Different values of i in $Backbone^i$ indicate different backbones.

5.4 Effect of the Camera Count

On PrismGaze, PrismFusion is evaluated with one to eight available cameras. All combinations are used for the one- and eight-camera settings; for each camera count from two to seven, eight valid subsets are sampled randomly.



(a) PrismGaze (b) ETH-XGaze Within Subject (c) ETH-XGaze Person Specific

Figure 3: Effect of camera count and composition. (a) Mean test error on PrismGaze across camera subsets; shading denotes one standard deviation. (b–c) ETH-XGaze errors under inner-to-outer and outer-to-inner camera-sampling orders for the within-subject and person-specific protocols. Lower values indicate better performance.

ETH-XGaze contains 10 inner-ring cameras and 8 outer-ring cameras. One sampling scheme draws up to 10 cameras from the inner ring and then adds outer-ring cameras; the other draws up to 8 cameras from the outer ring and then adds inner-ring cameras. Three configurations are sampled randomly for each camera count in each scheme.

On PrismGaze, error generally decreases as more cameras become available, although the trend is not monotonic (Fig. 3). On ETH-XGaze, inner-ring subsets perform better at small camera counts, while adding outer-ring cameras does not consistently reduce error. Conversely, outer-ring subsets improve after inner-ring cameras are added. Camera composition therefore matters alongside camera count.

5.5 Effect of Camera Placement

We analyze camera placement using single-camera results and two- and three-camera configurations drawn from the eight fixed positions; Sec. A.2 gives the configuration details.

Single-camera error ranges from 1.76 cm for Camera E to 3.50 cm for Camera NE (Table 4(a)), showing substantial variation across camera positions.

To test whether single-camera quality carries over to camera pairs, we rank the eight positions by single-camera error and compute the mean rank of each pair. This mean rank correlates with the ranking of the final two-camera error (Spearman $\rho = 0.553$, position-label permutation $p < 0.05$): pairs containing stronger individual cameras tend to have lower error.

Spatial spread is measured by inter-camera distance for camera pairs and by convex-hull area for three-camera configurations, with zero area assigned to collinear positions. Neither measure is significantly associated with test error (Spearman $p > 0.05$ in both cases).

These results show that camera quality is more informative than spatial spread alone when selecting camera combinations.

5.6 Error Analysis Across Target Locations

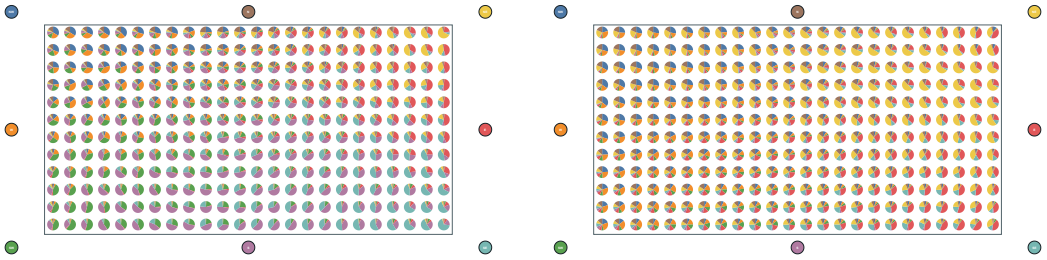
The display is divided into nine regions corresponding to the eight camera positions and the screen center. Each target is assigned to its nearest camera or the center point; Table 4(b) reports the resulting errors.

Screen region has a significant effect on prediction error (one-way ANOVA, $F = 19.40$, $p < 0.001$). Bonferroni-corrected Welch's t-tests show that Center has lower error than W, SW, E, S, SE, and NE, but does not differ significantly from NW or N. NW has lower error than SW, S, SE, and NE; E has lower error than S and SE; and N has lower error than SW, S, SE, and NE. The lowest mean errors occur at Center, NW, and N, while S, SE, and SW have the highest errors.

Figure 4 compares camera-level attention mass with and without Region Partition. In cross-attention, Z^{feature} provides the queries and values and Z^{weight} provides the keys. Attention dis-

(a) Single-camera error		(b) Error across target regions		
Position	**Error (cm)**	**Region**	**Targets**	**Error (cm)**
E	1.76	Center	72	0.73
SE	2.00	NW	18	0.78
S	2.37	N	36	0.78
W	2.51	E	36	0.89
NW	2.72	W	36	0.90
SW	3.07	NE	18	1.02
N	3.14	S	36	1.07
NE	3.50	SE	18	1.16
		SW	18	1.44

Table 4: Spatial error analysis on PrismGaze. (a) Single-camera error by camera position. (b) Multi-view gaze-estimation error by target region, reported in both pixels and centimeters. Rows in each panel are ordered by increasing centimeter error; lower values indicate better performance.



(a) With Region Partition. (b) Without Region Partition (ablation 1 in Table 3).

Figure 4: Target-wise cross-view attention with and without Region Partition. Each pie chart corresponds to one gaze target and shows the normalized attention mass assigned to the eight camera keys, with colors indicating camera position. Region Partition produces stronger spatial variation in camera attention across the screen.

tributions over key cameras are similar across query cameras, so we aggregate across queries using the number of valid samples as weights, average across layers, and normalize over key cameras.

Attention mass is negatively correlated with target-to-camera distance, with a stronger correlation under Region Partition ($\rho = -0.6763$) than without it ($\rho = -0.3730$). Thus, cameras closer to a target tend to receive more attention.

Attention mass is also negatively correlated with the corresponding single-camera error ($\rho = -0.345$, position-label permutation $p < 0.001$). Region Partition produces more spatially structured attention, assigning greater attention to cameras that are closer to the target and achieve lower single-view error.

5.7 Error Analysis in Real-world Scenario

The viewing-distance evaluation uses a separate group of 12 participants (6 female and 6 male; age 23.7 ± 3.75 years). Participants gaze at six peripheral calibration points and 44 points sampled independently for each participant on the same display as the main study. Data are collected at 60, 80, and 100 cm.

PrismFusion-Circle provides a geometric refinement by predicting the distance from the gaze point to each camera rather than $\hat{\mathbf{y}}$ directly. The predicted distances define circles centered at the eight camera locations, and nonlinear least squares recovers the point that best fits them. We compare PrismFusion, PrismFusion-Circle, and their averaged prediction. For the two-view setting, we report the camera pair with the lowest mean error across the three distances.

Calibration fits an affine transformation for each participant and distance using the six peripheral targets and applies it to the remaining 44 targets.

Method	60 cm	80 cm	100 cm
Tobii Eye Tracker	**1.38*****	**1.80*****	3.42
PrismFusion	9.62	10.98	12.31
PrismFusion-Circle	5.33	6.32	7.60
Average of two	5.58	6.47	8.03
PrismFusion (2 views)	6.11	5.89	7.93
PrismFusion calibrated	2.88	3.06	3.57
PrismFusion-Circle calibrated	2.43	2.76	**3.48**†
Average of two calibrated	2.10	2.29	**2.80****
PrismFusion (2 views, calibrated)	1.89	**1.87**†	**2.49****

Table 5: Mean gaze-estimation error in centimeters at viewing distances of 60, 80, and 100 cm (lower is better). The lower block applies a six-point affine calibration independently for each participant and distance. Symbols report Holm-corrected paired Wilcoxon comparisons with Tobii: *** denotes $p < 0.001$, ** denotes $p < 0.01$, and † denotes no significant difference.

At 60 and 80 cm, Tobii has lower error than all uncalibrated methods and most calibrated variants in Table 5. Calibrated two-view PrismFusion does not differ significantly from Tobii at 80 cm.

At 100 cm, calibrated PrismFusion-Circle does not differ significantly from Tobii, whereas the calibrated averaged prediction and calibrated two-view PrismFusion have lower error than Tobii (2.80 and 2.49 cm, respectively; $p < 0.01$). Calibration reduces the error of all PrismFusion variants at each distance.

6 Conclusion and Limitations

We presented PrismGaze, a multi-view dataset for screen-based gaze estimation that captures continuous head-pose variation while participants fixate on the same targets. We further proposed PrismFusion, a multi-view feature-fusion framework that uses region partitioning during training to encourage the integration of complementary evidence across camera views. Our evaluations across camera configurations, target locations, and viewing distances show that the main advantage of multi-view gaze estimation is its ability to compensate for unfavorable or poorly observed views using more informative viewpoints. The analysis further suggests that region partitioning promotes spatially structured attention: cameras closer to a target and with lower single-view error tend to receive greater attention during fusion. In addition, PrismFusion and its circle-based refinement remain effective under changes in viewing distance, especially when combined with lightweight calibration.

This work has several limitations. First, PrismGaze was collected in a controlled desktop setting with a limited number of participants, fixed camera placements, and relatively consistent illumination. Its results may therefore not fully generalize to substantially different devices, environments, demographics, or user behaviors. Second, participants removed glasses during the main data collection, so robustness to eyewear, severe occlusion, and other challenging visual conditions requires further study. Third, our method assumes synchronized multi-view inputs, which can increase hardware and deployment costs compared with single-camera solutions. Future work should investigate more diverse and longitudinal data collection, robustness to practical visual disturbances, flexible camera configurations, and efficient deployment with fewer or asynchronously available views.

Statement

AI Use Statement

In this work, we used generative AI tools to assist with the implementation of the experimental system, analysis and interpretation of experimental data, and preparation of visual materials, including tables and figures. We also used generative AI tools to assist with drafting and language editing of the manuscript. All AI-assisted outputs were reviewed, verified, and revised by the authors. The

authors take full responsibility for the final content of this work, including all text, claims, analyses, and visual artifacts.

Ethics Statement

This work involves the collection and planned release of synchronized facial images and gaze annotations, which may constitute sensitive biometric data. Participation was voluntary, and informed consent was obtained from all participants prior to data collection. We will release the dataset only under terms that prohibit attempts to identify participants or use the data for surveillance, discriminatory profiling, or other harmful applications.

Reproducibility Statement

To support reproducibility, we will release the PrismGaze dataset, source code, trained-model configurations, and scripts for preprocessing, training, and evaluation, subject to participant-consent and privacy requirements. The main paper and appendix describe the dataset protocol, subject-wise data splits, model design, and evaluation procedures. The released materials will include the configurations needed to reproduce the reported experiments and facilitate comparisons under different camera subsets, target locations, and viewing distances.

References

Yiwei Bao and Feng Lu. Unsupervised gaze representation learning from multi-view face images. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1419–1428. IEEE, 2024.

Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *2017 IEEE international conference on computer vision (ICCV)*, pp. 1021–1030. IEEE, 2017.

Jixu Chen and Qiang Ji. 3d gaze estimation with a single camera without ir illumination. In *2008 19th international conference on pattern recognition*, pp. 1–4. IEEE, 2008.

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

Yihua Cheng and Feng Lu. DVGaze: Dual-view gaze estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20632–20641, October 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Cheng_DVGaze_Dual-View_Gaze_Estimation_ICCV_2023_paper.html.

Anatolii Evdokimov, Catherine Finegan-Dollak, and Arryn Robbins. Weyeds: A desktop webcam dataset for gaze estimation. In *Proceedings of the 2024 Symposium on Eye Tracking Research and Applications*, pp. 1–2, 2024.

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

John Gideon, Shan Su, and Simon Stent. Unsupervised multi-view gaze representation learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4997–5005. IEEE, 2022.

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

Yoichiro Hisadome, Tianyi Wu, Jiawei Qin, and Yusuke Sugano. Rotation-constrained cross-view feature fusion for multi-view appearance-based gaze estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5985–5994, January 2024. URL https://openaccess.thecvf.com/content/WACV2024/html/Hisadome_Rotation-Constrained_Cross-View_Feature_Fusion_

for_Multi-View_Appearance-Based_Gaze_Estimation_WACV_2024_paper.html.

Harsimran Kaur, Swati Jindal, and Roberto Manduchi. Rethinking model-based gaze estimation. *Proceedings of the ACM on computer graphics and interactive techniques*, 5(2):1–17, 2022.

Taejun Kim, Vimal Mollyn, Riku Arakawa, and Chris Harrison. Hifigaze: Improving eye tracking accuracy using screen content knowledge. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2026.

Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2176–2184, June 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Krafka_Eye_Tracking_for_CVPR_2016_paper.html.

Dongze Lian, Lina Hu, Weixin Luo, Yanyu Xu, Lixin Duan, Jingyi Yu, and Shenghua Gao. Multiview multitask gaze estimation with deep convolutional neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 30(10):3010–3023, October 2019. doi: 10.1109/TNNLS.2018.2865525. URL https://pubmed.ncbi.nlm.nih.gov/30183647/.

Ruicong Liu and Feng Lu. Uvagaze: Unsupervised 1-to-2 views adaptation for gaze estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 3693–3701, 2024.

John Merchant, Richard Morrissette, and James L Porterfield. Remote measurement of eye direction allowing subject motion over one cubic foot of space. *IEEE transactions on biomedical engineering*, (4):309–317, 1974.

Carlos H Morimoto and Marcio RM Mimica. Eye gaze tracking techniques for interactive applications. *Computer vision and image understanding*, 98(1):4–24, 2005.

Seonwook Park, Emre Aksan, Xucong Zhang, and Otmar Hilliges. Towards end-to-end video-based eye-tracking. In *Computer Vision – ECCV 2020*, volume 12357 of *Lecture Notes in Computer Science*, pp. 747–763. Springer, 2020. doi: 10.1007/978-3-030-58610-2_44. URL https://www.ecva.net/papers/eccv_2020/papers_ECCV/html/1728_ECCV_2020_paper.php.

Jiawei Qin, Takuru Shimoyama, and Yusuke Sugano. Learning-by-novel-view-synthesis for full-face appearance-based 3d gaze estimation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4977–4987. IEEE, 2022.

Sheng-Wen Shih and Jin Liu. A novel approach to 3-d gaze tracking using stereo cameras. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1):234–245, 2004.

Yunjia Sun, Jiabei Zeng, Shiguang Shan, and Xilin Chen. Cross-encoder for unsupervised gaze representation learning. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3682–3691. IEEE, 2021.

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

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

Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. ETH-XGaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In *Computer Vision – ECCV 2020*, volume 12350 of *Lecture Notes in Computer Science*, pp. 365–381. Springer, 2020. doi: 10.1007/978-3-030-58558-7_22. URL https://www.ecva.net/papers/eccv_2020/papers_ECCV/html/3477_ECCV_2020_paper.php.

A Appendix

A.1 Off-Screen Target Ground Truth

Since both the side-view cameras and the display remain fixed throughout data collection, the mapping from screen coordinates to each side-view camera image is constant. For each side-view camera, we manually identify the four corners of the display and estimate a affine matrix between the screen coordinate system and the corresponding quadrilateral region in the camera image. The predefined 24×12 target grid is then projected through this affine matrix, providing an image-space ground-truth location for every on-screen fixation target.

Although the screen is visible in the side-view images, its content may reveal the currently displayed fixation target. To avoid this source of data leakage, we mask the entire screen region in the input images before they are provided to the model. This masking operation only modifies the model input and does not affect the ground-truth coordinates obtained from the geometric projection.

For off-screen targets, the physical objects remain at fixed locations throughout data collection. We manually annotate the center of each object once in each side-view camera image and use this location as its image-space ground truth. Participants are instructed to fixate on the designated object center during the corresponding trials.

A.2 Camera Configuration Analysis

The single-camera experiments cover all eight camera positions to evaluate the baseline performance of each individual viewpoint. The two-camera experiments include all 28 pairwise combinations of the eight positions. The three-camera experiments use 18 pre-selected configurations. These configurations were selected to represent geometrically distinct layouts, including relatively compact configurations and configurations with broader spatial spread. Since the three-camera configurations do not exhaust all possible combinations, the corresponding analyses should be interpreted as exploratory.

A.2.1 Single-Camera Baseline Performance

For the single-camera condition, each of the eight camera positions was trained and evaluated independently. The resulting test errors were used to rank individual camera positions. A lower rank indicates lower test error and therefore better single-camera performance.

A.2.2 Relationship Between Single-Camera Quality and Two-Camera Performance

To examine whether the quality of individual cameras transfers to two-camera configurations, each camera position was assigned a rank according to its single-camera test error. For a two-camera configuration consisting of cameras i and j, we defined its mean single-camera rank as

$$R_{i,j} = \frac{R_i + R_j}{2}, \tag{1}$$

where R_i and R_j are the single-camera error ranks of cameras i and j, respectively.

We then computed the Spearman correlation between $R_{i,j}$ and the rank of the final two-camera test error across all 28 pairwise configurations. A positive correlation indicates that pairs composed of better-ranked individual cameras tend to achieve better two-camera performance.

Because the 28 two-camera configurations are constructed from only eight camera positions, they are not independent observations: each camera appears in multiple pairs, causing different configurations to share the same camera-level information. Consequently, the standard significance test for Spearman correlation, which treats pairwise observations as independent, may not provide an appropriate null distribution. We therefore use a position-label permutation test that preserves the original two-camera combinations and their test errors while breaking the association between camera position and single-camera performance.

Specifically, under the null hypothesis that single-camera performance is unrelated to the performance of two-camera configurations, we randomly reassign the eight single-camera ranks to the eight camera positions. For each permutation, we recompute the mean single-camera rank $R_{i,j}$ for all 28 camera pairs and calculate its Spearman correlation with the observed two-camera error ranking. Repeating this procedure produces a null distribution of correlation coefficients that retains the dependency structure induced by shared camera positions. The two-sided permutation p value is computed as

$$p = \frac{1 + \sum_{b=1}^{B} \mathbb{I}\left(|\rho_b| \geq |\rho_{\text{obs}}|\right)}{B + 1}, \tag{2}$$

where ρ_{obs} is the observed correlation, ρ_b is the correlation obtained from the b-th permutation, and B is the number of permutations.

A.2.3 Inter-Camera Distance in Two-Camera Configurations

For each two-camera configuration (i, j), the spatial separation between cameras was quantified by the Euclidean distance

$$D_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{3}$$

We evaluated the relationship between $D_{i,j}$ and two-camera test error using both Spearman correlation and Pearson correlation. Spearman correlation tests for a monotonic relationship, whereas Pearson correlation tests for a linear relationship. Statistical significance was evaluated by randomly permuting the test errors across the 28 two-camera configurations while holding the distance values fixed.

A.2.4 Convex-Hull Area in Three-Camera Configurations

For each three-camera configuration, spatial coverage is quantified by the convex-hull area of the camera positions, with zero assigned to collinear layouts:

$$A_{\text{hull}} = \text{Area}\left[\text{ConvexHull}\left(\{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}\right)\right]. \tag{4}$$

Spearman and Pearson correlations relate A_{hull} to test error; significance is obtained by permuting errors across the 18 configurations while keeping the convex-hull areas fixed.