Multi-view Gaze Target Estimation

Supplementary Material

Abstract

In this supplementary material, we provide additional information for the implementation details (S1) and statistics of the MVGT dataset (S2). We extend our model to use more than 2 views (S3). We provide detailed information for the absolute depth computation (S4) and camera calibration procedure (S5). We show the angular errors of the predicted gaze vectors of our model (S6). We provide further analyses of the HIA (S7), UGS (S8), and ESA (S9) modules . We analyze the sensitivity of the model to the camera parameters (S10) and analyzed the complexity and costs of our model (S11). We show the performance of training our model with the same learning rate when evaluating on different scenes (S12). We show the performance of a reproduced baseline model on the GazeFollow dataset (S13). We provide additional discussions of the applicability, generalization, and limitations of our dataset and model (S14).

S1. Implementation Details

In the overall loss computation, we used $\alpha = 10.0$ and $\lambda = 0.1$. As mentioned in the main text, we performed leave-one-scene-out cross-validation in our experiments. β was set as 0.05 when leaving out the commons room scene for evaluation, and 0.3 when evaluating other scenes. We used a learning rate of 5×10^{-5} and a batch size of 40 when pretraining the single-view version of our model on GazeFollow [8]. When fine-tuning the model on MVGT, we first train the gaze estimator with a learning rate of 1×10^{-4} with a batch size of 60. Then we fine-tune the full model using a batch size of 40 pairs of views. The learning rate is 2.5×10^{-6} when evaluating the lab and store scenes, 2.5×10^{-5} when evaluating the commons scene, and 1×10^{-5} for the kitchen scene. For the cross-view task, due to the relatively small number of samples, we used a learning rate 1×10^{-7} for the research lab and commons room scenes, and 1×10^{-8} for the store and kitchen scenes . To obtain the eye location of the subject when computing the field-of-view (FoV) heatmap, we apply a face keypoint estimator [2] to the head crop of the subject and locate the eye keypoint location. If no eye is detected (e.g., the person is looking away from the camera), we choose the center point of the head bounding box as the eye location.

S2. Additional Dataset Information

Here we provide more detailed statistics on the division of of the dataset regarding the head and gaze target visibilities in the reference view. As shown in Tab.S1, we divide all the 68,430 camera view pairs into 4×4 cells according to the head and face visibilities of the primary and reference view. In Tab.1 in the main paper, we compared with single-view gaze target estimation (GTE) baselines, and evaluated the models on the primary view images. The cells shaded in green are the samples used in this scenario, as the primary view images must be applicable for GTE with themselves (head is visible). Within those samples, the ones with the target visible for the primary view are used for the GTE task, while all the samples shaded in green are used for the in/out classification task. On the other hand, the cells shaded in red are the pairs that we used for cross-view GTE and in/out classification, in which the head is only visible in the reference view. The rest of the pairs (white) are not used in any experiments, as neither single-view nor multi-view GTE methods are applicable due to the head not being visible in either view.

Reference View		Head Vis.		Head Not Vis.	
Primary View		Target Vis.	Target Not Vis.	Target Vis.	Target Not Vis.
Head Vis.	Target Vis.	11216	12314	660	2485
	Target Not Vis.	12314	13164	1254	2738
Head Not Vis.	Target Vis.	660	1254	526	1210
	Target Not Vis.	2485	2738	1210	2202

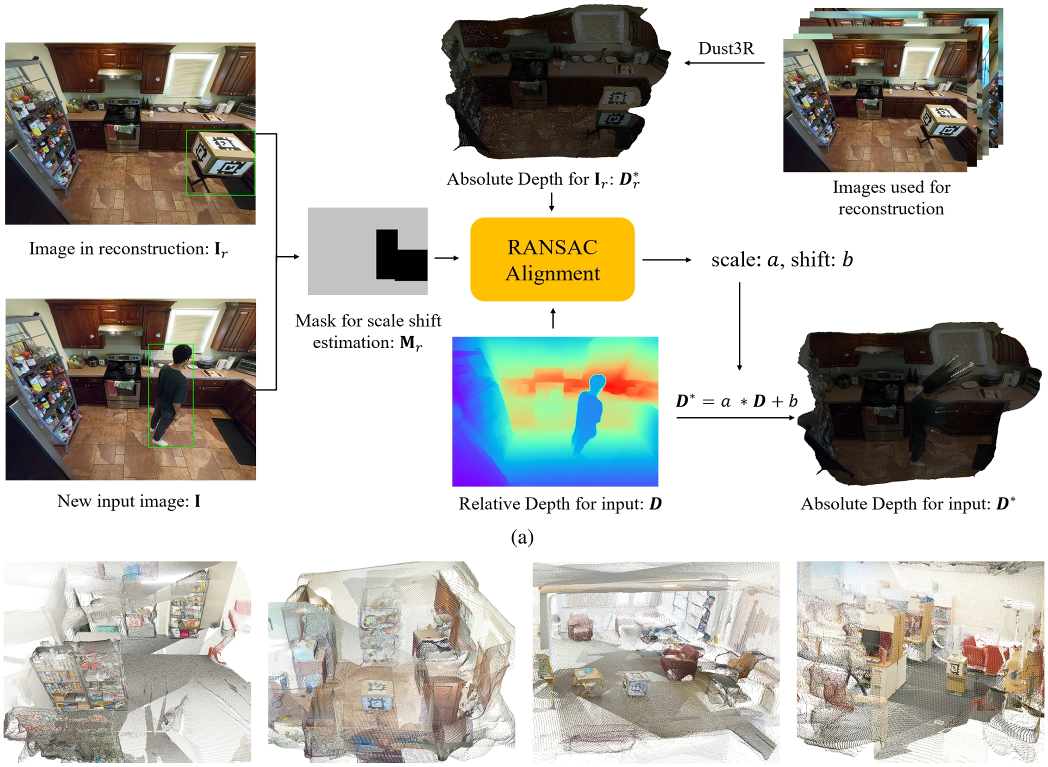
Table S1. Statistics of the camera view pairs in MVGT dataset regarding head and face visibilities in both the primary and reference views. Cells shaded in green are the camera view pairs used in comparison with single-view GTE methods. Cells shaded in red correspond to the pairs used in the cross-view GTE experiment.

S3. Extension to More Camera Views

Method	Views	Dist. ↓	AP ↑
Ours-Single	1	0.150	0.868
Ours	2	0.130	0.894
Ours	4	0.121	0.897
Ours	6	**0.118**	**0.898**

Table S2. Results of extending to more camera views. Our method shows greater improvements when using more camera views.

In this section, we extend our method to more than 2 camera views with a simple strategy by using multiple view


Dust3R
Absolute Depth for $\mathbf{I}_r$: $\boldsymbol{D}_r^*$
Images used for reconstruction
Image in reconstruction: $\mathbf{I}_r$
Mask for scale shift estimation: $\mathbf{M}_r$
RANSAC Alignment
scale: a, shift: b
$\boldsymbol{D}^* = a * \boldsymbol{D} + b$
New input image: $\mathbf{I}$
Relative Depth for input: $\boldsymbol{D}$
Absolute Depth for input: $\boldsymbol{D}^*$

(a)

(b)

Figure S1. (a). The procedure of absolute depth estimation. We first reconstruct the scene and obtain the absolute depth for all views using a set of images from 6 cameras (e.g., calibration images). For a new input image, we estimate the scale and shift between the monocular depth map and the absolute depth obtained from reconstruction using RANSAC, by masking out the person bounding box location along with the calibration cube location in the reconstruction image. We use the estimated scale and shift values to obtain the absolute depth for the new input image. (b) Example point clouds of the entire scenes reconstructed by Dust3R [12] using 6 images from 6 cameras.

pairs with the same primary view image simultaneously. E.g., when using 4 camera views, we use 3 view pairs where the same primary view (e.g. Camera1) is paired with 3 different reference view images (e.g., Cam2, Cam3, Cam4). For each pair with index i, we obtain an uncertainty score σ^i from the view with the lower σ value (i.e., the view selected in UGS). From all 3 pairs with their obtained uncertainty scores $\sigma^1, \sigma^2, \sigma^3$, we choose the output from the pair with the lowest σ value as the prediction for the primary view. This enables our method to leverage more camera views without any additional training.

Tab.S2 shows the results. Unlike Tab.1 in the main paper, here we investigate the effect of the number of camera views used without considering the head and gaze target visibility of the reference views. In this way, a primary view image is paired with each of the 5 other camera views and forms 5 pairs. In evaluation, for each primary view image, we randomly select 1 pair when using 2 views, 3 pairs when using 4 views, and all 5 pairs when using 6 views. When using 2 or 4 camera views, we run the model 5 times and report the average performance. Although our method with two views already outperforms the single-view baseline significantly, using more views shows even further improvement.

S4. Absolute Depth Computation

In this section, we describe our procedure for estimating the absolute depth in cross-view GTE. As shown in Fig.S1a, we first reconstruct the 3D scene using a set of images from all 6 cameras (e.g. images used in calibration) and obtain the absolute depth $\boldsymbol{D}_r^*$ for each camera view. As we mentioned in the main paper, by inputting the camera parameters calibrated in real-world metrics, Dust3R can generate depth estimations that are very close to the absolute depth values by optimizing a reconstruction loss. After the 3D reconstruction, when a new input image comes during training and evaluation, we estimate the scale and shift between its relative depth map $\boldsymbol{D}$ from a monocular depth estimation model and the absolute depth $\boldsymbol{D}_r^*$ obtained from reconstruction by masking out the area that changes with a mask $\mathbf{M}_r$. We use RANSAC to estimate the scale and shift for the input image with this mask $\mathbf{M}_r$:

$$a, b = \underset{\mathbf{M}_r^{(i,j)}==1}{RANSAC}(\boldsymbol{D}(i,j), \boldsymbol{D}_r^*(i,j)), \tag{1}$$

With the estimated scale and shift, we can obtain the absolute depth $\boldsymbol{D}^*$ of the new input image: $\boldsymbol{D}^* = a * \boldsymbol{D} + b$. Note that after reconstruction, we do not require additional camera views to be available for input. We run this procedure for both the primary and reference views. With the absolute depth from both views obtained, we can obtain the real 3D eye location in the reference view, and transform the location to the primary view's coordinate system using the extrinsic parameters. In this way, the FoV heatmap for the primary view can be obtained in cross-view GTE settings, even when the person is not visible in the primary view.

S5. Camera Calibration Details

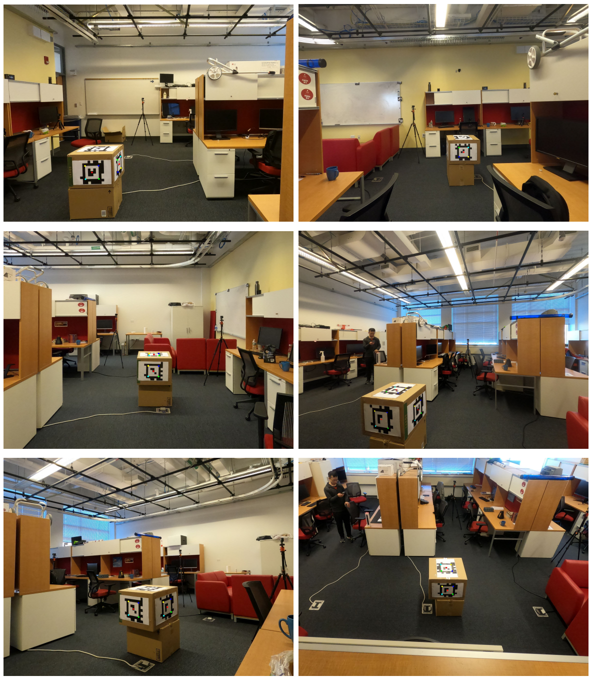


Figure S2. Protocol for calibrating extrinsic camera parameters. We use a cube stuck with AprilTag patterns to calibrate all cameras' extrinsic parameters. The automatically detected corners for the patterns are visualized. We measured the 3D physical locations of these corners before calibration.

In this section, we provide more details of our camera calibration procedure. We used chessboard patterns for calibrating the intrinsic parameters, which is a common protocol for intrinsic parameter calibration. For calibrating the extrinsic camera parameters, we used a cube stuck with AprilTag patterns on different faces. As shown in Fig.S2, we first measure the 3D physical locations of all the AprilTag corners in the cube, and put the cube in a location where it can be observed by all cameras. The 2D locations of the corners can be automatically detected [7], and the extrinsic parameters can be obtained using PnP (Perspective-nPoints). By using this protocol, the extrinsic parameters can be obtained with just one set of images taken. We calibrate the extrinsic parameters before the data collection session of each subject. If no position is visible to all cameras without occlusion, calibration is performed twice using a common camera as the shared coordinate system.

S6. Analyses of Predicted Gaze Vectors

In this section, we specifically investigate the angular errors of the predicted gaze vectors in our model under different ablation conditions. The ground truth gaze vectors are obtained from the 3D eye locations and gaze target locations computed from triangulation when they appear in multiple camera views. This provides more accurate ground truths than the "pseudo" gaze vectors used in training. Tab.S3 shows the results. Training with the uncertainty loss results in an overall improvement in predicted gaze vectors. The HIA leads to a significant improvement when the head is visible, showing its effectiveness in aggregating head information from another view. It also shows a small improvement when the head is not visible. We hypothesize that this is because the model benefits from the overall multi-view training. The UGS module shows improvement when the reference view includes the head but not the gaze target, where the subject is typically facing the camera with a clearly visible face and is selected as the more reliable gaze vector. The ESA module is responsible for aggregating scene information, and does not show a large change in gaze vector prediction.

σ	HIA	UGS	ESA	Head Vis.		Head Not Vis.	
				Target Vis.	Target Not Vis.	Target Vis.	Target Not Vis.
				Ang. ↓	Ang. ↓	Ang. ↓	Ang. ↓
				28.71	28.93	29.15	29.73
✓				26.94	27.28	27.69	28.22
✓	✓			21.20	21.32	25.22	26.40
✓	✓	✓		<u>20.50</u>	**19.72**	<u>24.87</u>	**25.78**
✓	✓	✓	✓	**20.47**	<u>20.02</u>	**24.68**	<u>25.84</u>

Table S3. Angular Errors of predicted gaze vectors of our method under different ablation conditions.

S7. Analyses of HIA Module

In this section, we provide more detailed analyses of the HIA module. We first analyzed the effects of the two additional inputs of HIA: the head crop image in the other view, and the relative rotations between views computed from the extrinsic parameters. Tab.S4 shows the results. Here we treat our method only with the HIA module (without UGS and ESA) as the base model to avoid the potential influence of other components. In the 2nd row, we discard the head image input in the reference view, by inputting zero-padded tensors to the HIA module in all cases. It shows a large drop in all metrics when the head is visible, demonstrating that HIA effectively leverages the head appearances from the other view to enhance the embeddings. The performance does not change when the head is not visible, as the input is the same. In the first row, we trained the model with HIA by removing the input camera parameters (relative rotations). This also results in a significant drop when the head is visible. This suggests that without the geometric relationship between views, the model cannot learn how to effectively uses the head appearance from the other view. There is also a large drop in AP when the head is not visible in the reference view. This shows that with the input camera parameters, the head embedding is enhanced with 3D geometric-aware information and benefits the performance of in/out prediction when input to the in/out prediction head, as shown in Fig.4 in the main paper.

Method	Head Vis.				Head Not Vis.			
	Target Vis.		Target Not Vis.		Target Vis.		Target Not Vis.	
	Dist. ↓	AP ↑	Dist. ↓	AP ↑	Dist. ↓	AP ↑	Dist. ↓	AP ↑
No Cam.	0.150	0.880	0.152	0.875	0.175	0.776	**0.154**	0.847
No Head	0.149	0.876	0.155	0.886	**0.174**	**0.821**	0.155	**0.873**
Ours-HIA	**0.135**	**0.896**	**0.133**	**0.897**	**0.174**	**0.821**	0.155	**0.873**

Table S4. Analysis of the HIA module. We experimented by discarding the head crop input or training without inputting camera parameters, using our model with only HIA as the base model.

We also explored alternative strategies for aggregating head features and training the gaze estimator. In Tab. S5 (Row 1), we replaced concatenation with multiplication when including the camera rotation matrix in HIA's cross-attention, which led to worse performance, especially when the head is not visible in the reference view. In Row 2, fixing the gaze backbone during training also resulted in significantly degraded performance.

Method	Head Vis.				Head Not Vis.			
	Target Vis.		Target Not Vis.		Target Vis.		Target Not Vis.	
	Dist. ↓	AP ↑	Dist. ↓	AP ↑	Dist. ↓	AP ↑	Dist. ↓	AP ↑
Ours-Mul	0.131	0.903	0.124	0.904	0.172	0.801	0.160	0.860
Ours-Fix	0.145	0.894	0.139	0.897	0.193	0.805	0.171	0.841
Ours	**0.129**	**0.909**	**0.122**	**0.912**	**0.161**	**0.836**	**0.152**	**0.868**

Table S5. Results of using multiplication for the relative rotation (Row 1) and fixing the gaze backbone during training (Row 2).

S8. Analyses of UGS Module

In this section, we analyze the UGS module in detail by demonstrating the correlation between the error in gaze vector prediction and the predicted uncertainty score σ, the performance of UGS on samples with large errors in predicted gaze vectors, and showing the effect of the UGS module with some qualitative examples.

In Fig.S3 we visualize the average angular error of the predicted gaze vectors with their predicted uncertainty scores σ falling into different slots. About 93% of all samples have a predicted $\sigma < 0.2$. It can be seen a larger σ value corresponds to a larger error for the gaze vector prediction, supporting our motivation to select the view with a lower σ value in an input view pair in the UGS module.

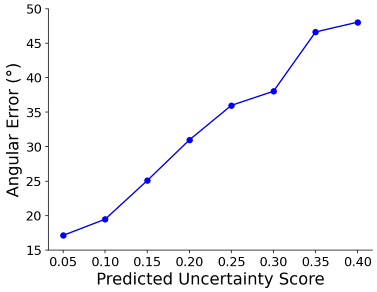


Figure S3. Average angular error for the predicted gaze vectors with their uncertainty scores σ falling into different slots. We divide the slot of σ by 0.05. The gaze vectors with larger predicted uncertainty scores tend to have larger angular errors.

We also demonstrated the effectiveness of the UGS module by operating the module on samples with large errors in predicted gaze vectors, of which the predicted gaze vectors have an angular error $> 30°$ before uncertainty-based selection. To show the effectiveness of UGS, the models were evaluated on all view pairs where the reference view contains the head of the subject. As shown Tab.S6, in addition to the substantial reduction in angular errors for gaze vectors after selection, the Dist. and Ap. metrics in GTE also exhibit notable improvements, highlighting its crucial impact on samples with large initial prediction errors.

Method	Dist. ↓	AP ↑	Ang. ↓
No selection	0.228	0.856	44.35°
Ours	**0.200**	**0.883**	**35.91°**

Table S6. Effect of the UGS module on samples with large errors in predicted gaze vectors in the primary view before selection.

Fig.S4 demonstrates the effect of the UGS module. By leveraging the reference view with a lower σ value, the unreliable gaze vector from the primary view will be replaced with a much more accurate gaze vector, and lead to an FoV heatmap with a much better quality.

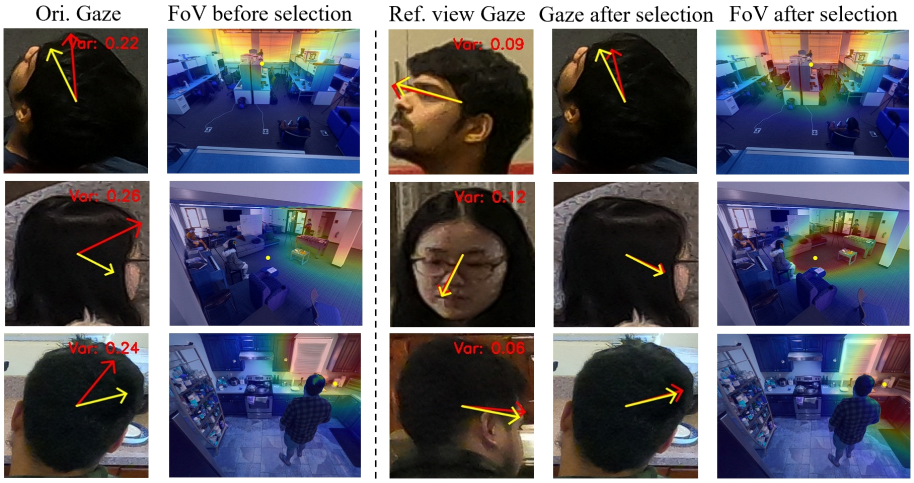


Figure S4. Qualitative examples for the UGS module output. The left side shows the gaze vectors and FoV heatmaps without using UGS, and the right shows the gaze vector and FoV heatmaps after selection and replacement in UGS. Red vectors correspond to the predicted gaze vectors while yellow ones are the ground truth. Leveraging the reference view with a more accurate gaze vector predicted and a lower uncertainty score, the UGS module can output a FoV heatmap with much better quality.

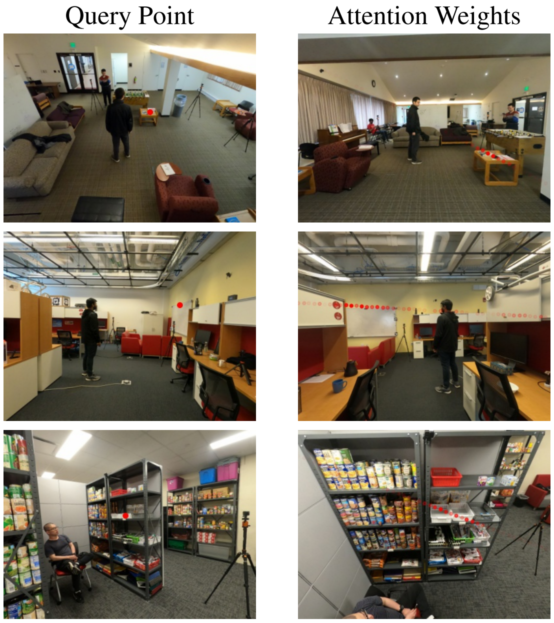


Figure S5. Visualizations of attention weights in the ESA module. The right column visualizes the attention weights between the query point in the left column and the sampled feature tokens along the epipolar line corresponding to the query in the reference view. Larger attention values are shown with higher intensity. The tokens located near the same location as the query point show the highest weights along the entire epipolar line.

S9. Analyses of ESA module

In this section, we provide more analyses of the ESA module, by showcasing its performance on the samples with occlusion, and visualizing the attention weights of the epipolar attention in the ESA module. We show the effectiveness of the ESA module by visualizing the epipolar attention weights. As explained in the main paper, each feature token in one view will be engaged in cross attention with feature tokens sampled along the epipolar line in the other view. In Fig.S5, we select a query feature location in the primary view and visualize the attention weights with the sampled feature tokens along the epipolar line in the reference view. We average the attention weights across all heads in the cross-attention module. It can be seen the attention weights are the highest for tokens located near the same location as the query point, demonstrating the effectiveness of the ESA module in aggregating useful scene background information from the other view.

On the other hand, we also perform an ablation study of the ESA module on samples which are occluded in the primary view, to investigate the effectiveness of epipolar attention in differentiating occluding objects by using information from another view. We manually annotated the locations of the occluded samples in the primary view by referring to the target locations (laser points) in other views, making up 1576 pairs of input views. Most of these targets are partially occluded by another object, or self-occluded, i.e., the laser point is located on the invisible side of the object. As the total number of occluded samples are relatively small and we assigned all occlusion samples to the "inside" class for the in/out task, we observed the model showing performance close to 1 in AP. Therefore, we just evaluated the Dist. metric for the GTE task.

Tab.S7 shows the results. Compared to the model without ESA module, the full model shows an obvious performance when the target is visible in the reference view. This supports our claim in the main paper that ESA provides complementary information on the potential gaze object from the reference view, especially in disambiguating the gaze target in case of occlusion in the primary view.

Method	Head Vis.		Head Not Vis.	
	Target Vis.	Target Not Vis.	Target Vis.	Target Not Vis.
	Dist. ↓	Dist. ↓	Dist. ↓	Dist. ↓
No ESA	0.161	**0.152**	0.171	0.131
Ours	**0.150**	0.155	**0.162**	**0.130**

Table S7. Ablation of ESA module on samples with occlusion. Without ESA, the model shows obvious drop in performance when the target is visible in the reference view.

S10. Sensitivity Analysis to Camera Parameters

In the main paper, we provide the results of our model with calibrated camera parameters. In this section, we analyze the sensitivities of our model to changes in camera parameters by considering potential errors and noise in camera calibration in real applications. We randomly jittered the intrinsic and extrinsic parameters by -5% ∼ 5%, and show the results of the multi-view and cross-view experiments in Tab.S8 and Tab.S9. As shown in both tables, the model only has a slight drop in performance with the jittered camera parameters in both general multi-view and cross-view tasks.

Method	Head Vis.				Head Not Vis.			
	Target Vis.		Target Not Vis.		Target Vis.		Target Not Vis.	
	Dist. ↓	AP ↑	Dist. ↓	AP ↑	Dist. ↓	AP ↑	Dist. ↓	AP ↑
Ours*	0.130	0.907	0.124	0.910	0.165	0.830	0.153	0.865
Ours	**0.129**	**0.909**	**0.122**	**0.912**	**0.161**	**0.836**	**0.152**	**0.868**

Table S8. Results of sensitivity analyses to camera parameters. Ours* shows the performance of the model with camera parameters jittered by -5% ∼ 5%, and Ours is the original model. Our model shows little drop in performance in all conditions.

Method	Dist. ↓	AP ↑
Ours*	0.190	0.817
Ours	**0.188**	**0.820**

Table S9. Results of sensitivity analyses to camera parameters for the cross-view task.

S11. Model Complexity and Cost

Tables S10 and S11 present the model size and inference speed on an RTX A5000 GPU for the multi-view and cross-view settings. Our model has a relatively larger number of parameters compared to single view baselines due to the use of transformer [4, 11, 13] as the encoder. However, our method only adds a very small number of parameters for multi-view processing compared to Ours-single, and it shows a large improvement in performance. Regarding inference speed, our method runs at 61.53 ms per image (16.25 FPS), which is acceptable considering the theoretical lower bound is 2 x due to processing two images. Despite inference speed not being a primary design goal, we believe real-time performance is achievable with techniques like quantization.

Methods	Params.	Runtime (ms)	Dist. ↓ (Head Visible)
Chong[3]	61.5M	14.92	0.158
Miao[6]	61.7M	15.25	0.141
Tafasca*[10]	21.7M	16.74	0.148
Ours-single	101.9M	20.63	0.150
Ours	107.6M	61.53	**0.125**

Table S10. # Parameters and runtime of multi-view models.

Methods	Params.	Runtime (ms)	Dist↓
DeepGazeIIE[5]	104.1M	248.76	0.248
Recasens[9]	189.3M	10.48	0.271
Ours	108.4M	62.21	**0.188**

Table S11. # Parameters and runtime of cross-view models

S12. Training with the same learning rate

In the experiments, we trained the model with different learning rates when evaluating on different scenes in leave-one-scene-out cross validation. We show the results when training the model with the same learning rate in the general multi-view (Tab.S12) and cross-view (Tab.S13) tasks with a learning rate of 2.5×10^{-6} and 1×10^{-7} respectively. Even when training with the same learning rate without tuning for each scene specifically, the model only shows a small drop in performance in all conditions.

Method	Head Vis.				Head Not Vis.			
	Target Vis.		Target Not Vis.		Target Vis.		Target Not Vis.	
	Dist. ↓	AP ↑	Dist. ↓	AP ↑	Dist. ↓	AP ↑	Dist. ↓	AP ↑
Ours†	0.134	0.905	0.125	0.910	0.167	0.823	0.159	0.860
Ours	**0.129**	**0.909**	**0.122**	**0.912**	**0.161**	**0.836**	**0.152**	**0.868**

Table S12. Results of training with the same learning rate in the general multi-view task. Ours† shows the results of training the models with the same learning rate.

Method	Dist. ↓	AP ↑
Ours†	0.199	0.814
Ours	**0.188**	**0.820**

Table S13. Results of training with the same learning rate in the cross-view task.

S13. Reproduced results of Tafasca et. al.

In Tab.S14, we show the results in the paper and our reimplemented numbers for Tafasca et.al [10] on the GazeFollow dataset [8]. We can achieve almost the same performance. In Tab.1 in the main paper, the model trained on GazeFollow is fine-tuned on our MVGT dataset for evaluation.

Method	AUC ↑	Avg. Dist. ↓	Min. Dist ↓
Tafasca [10]	0.936	0.125	0.064
Tafasca*	0.935	0.124	0.063

Table S14. Results of the numbers in the paper and the reproduced results on the GazeFollow dataset for Tafasca et.al [10]. * indicates results for the reimplemented model.

S14. Discussions and Limitations

In this section, we discuss the applicability and generalization of our dataset and method. Although the MVGT dataset is not as large scale as most available GTE datasets, it is the first GTE dataset that includes valuable multi-view scene/subject information, with calibrated camera parameters and precise gaze target annotations. Furthermore, we also introduce a well-defined, non-intrusive data collection protocol for gathering GTE data with accurate annotations, which is easily applicable to new scenes with multi-view setups, and allows for potential scaling up of the dataset. We hope our dataset will inspire broader community contributions to multi-view GTE.

For handling different scenarios in multi-view GTE, we proposed two networks in the general multi-view and cross-view task scenarios. The first model is applicable without any assumptions; the second is modified from the first for cross-view GTE, but can only be used when 3D scene reconstruction is available before GTE. In real applications, when a 3D reconstruction exists, both models can be used: the first model is used when the head is in the primary view, and the second otherwise. If not, users can still use the first, which outperforms single-view methods significantly.

Regarding the generalization of our method, we demonstrate it in the main paper when evaluated on test scenes not seen in training. In Fig.S6, we also apply our models to some example images in the Shelf dataset [1], a multi-view human pose estimation dataset distinct from our dataset. Although no gaze target annotation in available in the dataset, it can be seen from the supplementary view (rightmost column) in the figure that our models predict reasonable locations in both general multi-view and cross-view settings by using the camera parameters provided in the dataset.

While our method is effective in multi-view and cross-view GTE, it still has limitations. The reliance on explicit camera parameters and 3D scene reconstruction in cross-view GTE may limit its applicability in real-world scenarios. Future work could explore learning geometric relationships without camera parameters or performing cross-view GTE without 3D scene reconstruction.

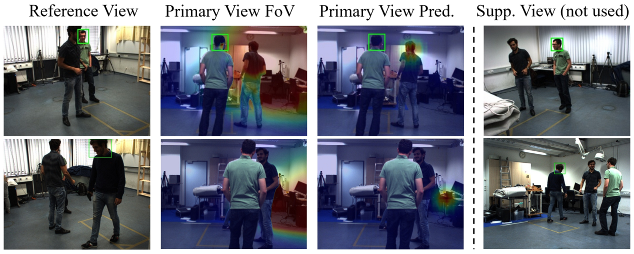


Figure S6. Our model evaluated on Shelf dataset. Output is shown for the person with an overlayed head box in both general multi-view (Row1) and cross-view (Row2) settings. The rightmost column is just for showing the potential target, and not used as input.

References

[1] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3d pictorial structures for multiple human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1669–1676, 2014. 7

[2] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017. 1

[3] Eunji Chong, Yongxin Wang, Nataniel Ruiz, and James M Rehg. Detecting attended visual targets in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5396–5406, 2020. 6

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of International Conference on Learning and Representation*, 2021. 6

[5] Akis Linardos, Matthias Kümmerer, Ori Press, and Matthias Bethge. Deepgaze iie: Calibrated prediction in and out-of-domain for state-of-the-art saliency modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12919–12928, 2021. 6

[6] Qiaomu Miao, Minh Hoai, and Dimitris Samaras. Patch-level gaze distribution prediction for gaze following. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 880–889, 2023. 6

[7] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *Proceedings of the IEEE Conference Robotics and Automation*, pages 3400–3407. IEEE, 2011. 3

[8] Adria Recasens, Aditya Khosla, Carl Vondrick, and Antonio Torralba. Where are they looking? In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2015. 1, 7

[9] Adria Recasens, Carl Vondrick, Aditya Khosla, and Antonio Torralba. Following gaze in video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1435–1443, 2017. 6

[10] Samy Tafasca, Anshul Gupta, and Jean-Marc Odobez. Childplay: A new benchmark for understanding children's gaze behaviour. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20935–20946, 2023. 6, 7

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, **L**ukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 6

[12] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 2

[13] Siqiao Zhao, Zhikang Dong, Zeyu Cao, and Raphael Douady. Hedge fund portfolio construction using polymodel theory and itransformer. *arXiv preprint arXiv:2408.03320*, 2024. 6