GA3CE: Unconstrained 3D Gaze Estimation with Gaze-Aware 3D Context Encoding Supplementary Material

Yuki Kawana Shintaro Shiba Quan Kong Norimasa Kobori
Woven by Toyota

A. egocentrict transformation

A.1. Pose normalization

In this section, we describe the computation of $\mathbf{t}_{\text{pose}}$ and s, which are used to normalize the 3D pose P_{pose} in the egocentric transformation introduced in Sec. 3.2. For 3D pose estimation, we use the off-the-shelf estimator from [20], which outputs 3D poses in the *smpl+head_30* format, comprising 30 keypoints representing the body and head.

The head position, $\mathbf{t}_{\text{pose}}$, is defined as the average 3D position of the head keypoints (indices 24 to 29). The scale s, representing the inverse head size, is computed as the reciprocal of the L2 norm between the 24th and 28th keypoints, which defines the head's width.

To simplify the representation, we subsample the keypoints using the indices $\{0, 4\text{–}8, 12, 18\text{–}21, 24, 26, 28, 29\}$, resulting in a reduced set of $N_{\text{pose}} = 15$ keypoints, as defined in Sec. 3.1.

A.2. Details on rotation alignment

Human vision studies [3, 4, 13] have shown that gaze fixations often concentrate near the center of the visual field, with distance information playing a critical role in gaze saliency prediction, particularly from a first-person perspective. This suggests that both the direction and distance of objects relative to the subject are key factors in estimating gaze direction. For example, individuals tend to focus more on nearby objects than distant ones, even when both lie in the same direction. Likewise, objects near the visual center attract more attention than those in the periphery, even if the latter are physically closer.

Furthermore, learning spatial relationships in a normalized space helps simplify the complex patterns of poses, object positions, and their interrelations. To this end, we normalize 2D observations into an egocentric view as the 3D context representations, as described in Sec. 3.1.

We assume that surveillance or monitoring cameras have their x-axis aligned with the horizon, the y-axis pointing downward (though not necessarily perpendicular), and the z-axis extending forward from the camera center, consistent with typical egocentric human perspectives. This assumption generally holds for common 2-DoF fixed cameras mounted on flat ceilings or walls. Even if the camera is tilted due to roll (rotation about the z-axis), this can be corrected using calibrated camera poses or scene cues such as vanishing points [9, 16].

As a result, any rotation about the camera's z-axis within the camera coordinate system, which rotates the x-y plane of the view, disrupts the consistent alignment of the camera view that maintains the horizon in the view as horizontal. Our goal is to normalize 3D context representations to this aligned view. During egocentric transformation, we apply rotation that maintain the consistency of the x-y plane w.r.t. the horizon, by avoiding rotations around the z-axis.

A.3. Axis-angle rotation alignment

An axis-angle representation is defined by a rotation axis $\mathbf{a} \in \mathbb{S}^2$ and a rotation angle $\eta \in \mathbb{R}$. To align the view direction $\mathbf{v} \in \mathbb{S}^2$ with a fixed direction $\mathbf{z} = [0, 0, 1]$, the axis

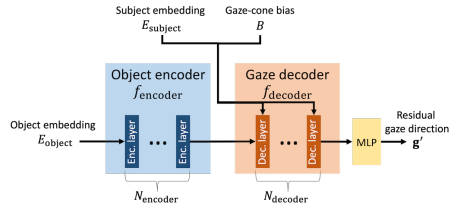

Figure 1. Architecture of the 3D gaze transformer. *Enc. layer* and *Dec. layer* refer to the transformer encoder and decoder layers, respectively.

Method	2D gaze	AUC ↑	L2 Dist. ↓	3D Dist. ↓	3D MAE ↓
GFIE [6]	✓	0.965	0.065	0.311	17.7
GFIE [6] + 3D	✓	0.978	0.062	0.341	16.4
GFIE [6] + ViTGaze [21]	✓	0.965	**0.054**	0.32	17.9
GFIE [6] + 3D + ViTGaze [21]	✓	0.978	**0.054**	0.30	16.6
Ours		-	-	-	11.1
Ours + GFM [6]	✓	**0.987**	0.067	**0.260**	**10.6**

Table 1. Quantitative results on the GFIE dataset [6]. *ViTGaze* refers to replacing the baseline GFIE's 2D gaze-following module from [21].

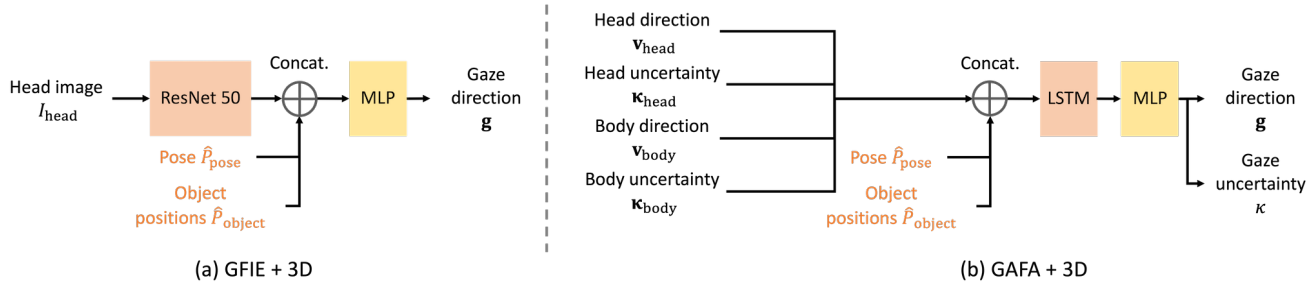

Figure 2. Illustration of the modifications to the baseline architectures for incorporating 3D context in (a) GFIE [6] and (b) GAFA [14].

Figure 3. Additional qualitative results on the GFIE dataset [6]. Red, green, and blue arrows indicate the ground truth, Ours, and the baseline GFIE [6], respectively.

Method	Input	2D gaze	AUC ↑	L2 Dist. ↓	3D Dist. ↓	3D MAE ↓
Random			0.469	0.758	1.910	70.3
Center			0.456	0.706	1.280	75.9
Rt-Gene [5]	H		0.463	0.492	0.483	26.5
Gaze360 [7]	H		0.463	0.474	0.427	20.6
GazeFollow [19]	SD	✓	0.862	0.196	1.030	44.1
Lian [10]	SD	✓	0.871	0.180	0.813	34.8
Chong [2]	SD	✓	0.891	0.152	0.812	31.9
$\text{GFIE}_{\text{head}}$ [6]	H		-	-	-	27.3
GFIE [6]	SD	✓	0.921	0.114	0.365	19.8
GFIE [6] + 3D	SD	✓	**0.933**	**0.094**	0.365	17.8
Ours*	SD		-	-	-	24.5
Ours	SD		-	-	-	25.2
Ours + GFM [6]	SD	✓	0.921	0.094	0.314	15.8
Ours + GFM†[6]	SD	✓	**0.933**	**0.094**	**0.243**	**14.6**

Table 2. Quantitative results on the CAD-120 dataset [8], with baselines from [6], are presented. Input modalities are denoted as follows: H = head image only; SD = scene image with depth map. A check mark indicates that the method requires 2D gaze following; otherwise, it directly estimates the 3D gaze direction. *GFM* refers to the gaze following modules from [6]. $\text{GFIE}_{\text{head}}$ uses only the head image as input without GFM [6]. Ours* uses a depth map from the zero-shot estimator [1]. GFM† uses the same module trained for GFIE + 3D, as detailed in Supp. B.2.

$\mathbf{a}$ and angle η are given by:

$$\mathbf{a} = \frac{\mathbf{v} \times \mathbf{z}}{\|\mathbf{v} \times \mathbf{z}\|}, \qquad \eta = \arccos(\mathbf{v}^T \mathbf{z}). \tag{1}$$

To examine how the rotation in Eq. (1) affects rotation around the z-axis, we convert it to intrinsic Euler angles in x-y-z order, denoted as $\{\theta, \phi, \psi\}$. The corresponding rotation matrix is expressed as $R_{\text{euler}} = R_x(\theta)R_y(\phi)R_z(\psi)$.

Using Rodrigues' rotation formula, the Euler angle ψ, representing the rotation around the z-axis, is computed as:

$$\psi = \text{arctan2}(c_\eta + a_x^2(1 - c_\eta), a_x a_y (1 - c_\eta) - a_z s_\eta), \tag{2}$$

where $\mathbf{a} = [a_x, a_y, a_z]$, $c_\eta = \cos\eta$, and $s_\eta = \sin\eta$. Unless $a_x a_y(1-c_\eta) - a_z s_\eta$ equals zero, the rotation defined by Eq. (1) includes a nonzero component around the z-axis.

A.4. Analytical solution for cyclotorsion rotation

Our objective is to keep ψ at zero and determine the angles θ and ϕ that satisfy the constraint on the view direction $\mathbf{v} = [v_x, v_y, v_z]$ relative to $\mathbf{z} = [0, 0, 1]$, as described in Eq. (1):

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ \sin\theta\sin\phi & \cos\theta & -\sin\theta\cos\phi \\ -\cos\theta\sin\phi & \sin\theta & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}. \tag{3}$$

Then, ϕ is defined as:

$$\phi = \text{arctan2}(-v_x, v_z) \tag{4}$$

Given ϕ, θ is defined as:

$$\theta = \text{arctan2}(v_y, v_z \cos\phi - v_x \sin\phi) \tag{5}$$

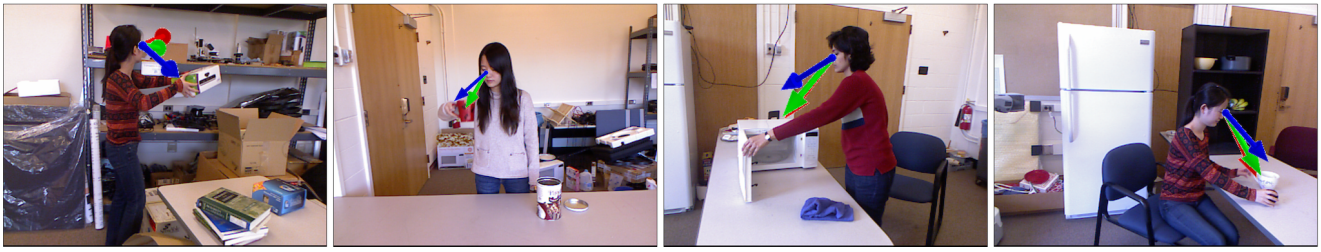

Figure 4. Additional qualitative results on the CAD-120 dataset [8]. Red, green, and blue arrows represent the ground truth, Ours + GFM [6], and the baseline GFIE [6], respectively.

Figure 5. Additional qualitative results on the GAFA dataset [14]. Red, green, and blue arrows represent the ground truth, Ours, and the baseline GAFA [14], respectively.

B. Additional implementation details

B.1. Details on positional encoding

We present the formulation and implementation details of the positional encoding in this section.

Since positional encoding is applied independently to each point in the normalized body keypoints P'_{pose} and normalized object positions P'_{object}, we omit N_{pose} and N_{object} in the following formulations for simplicity.

The standard positional encoding [23] for the view direction $\mathbf{v}'$ is defined as:

$$\gamma_{\text{view}} : \mathbb{S}^2 \to \mathbb{R}^{C_{\text{gaze}}}. \tag{6}$$

The D^3 positional encoding $\tilde{\gamma}_{\text{pose}}$ for a point in P'_{pose}, along with the standard positional encodings $\gamma^{\text{dir}}_{\text{pose}}$ and $\gamma^{\text{dist}}_{\text{pose}}$ for its direction and distance components, are defined as:

$$\begin{aligned} \tilde{\gamma}_{\text{pose}} &: \mathbb{R}^3 \to \mathbb{R}^{C_{\text{keypoint}}} \\ \gamma^{\text{dir}}_{\text{pose}} &: \mathbb{S}^2 \to \mathbb{R}^{C^{\text{dir}}_{\text{keypoint}}} \\ \gamma^{\text{dist}}_{\text{pose}} &: \mathbb{R} \to \mathbb{R}^{C^{\text{dist}}_{\text{keypoint}}} \end{aligned} \tag{7}$$

where $C^{\text{dir}}_{\text{keypoint}} = 6$, $C^{\text{dist}}_{\text{keypoint}} = 4$, and $C_{\text{keypoint}} = C^{\text{dir}}_{\text{keypoint}} + C^{\text{dist}}_{\text{keypoint}} = 10$.

Similarly, the D^3 positional encoding $\tilde{\gamma}_{\text{object}}$ for a point in P'_{object}, along with the standard positional encodings $\gamma^{\text{dir}}_{\text{object}}$ and $\gamma^{\text{dist}}_{\text{object}}$ for direction and distance, are defined as:

$$\begin{aligned} \tilde{\gamma}_{\text{object}} &: \mathbb{R}^3 \to \mathbb{R}^{C_{\text{latent}}} \\ \gamma^{\text{dir}}_{\text{object}} &: \mathbb{S}^2 \to \mathbb{R}^{C^{\text{dir}}_{\text{latent}}} \\ \gamma^{\text{dist}}_{\text{object}} &: \mathbb{R} \to \mathbb{R}^{C^{\text{dist}}_{\text{latent}}} \end{aligned} \tag{8}$$

where $C^{\text{dir}}_{\text{latent}} = 128$, $C^{\text{dist}}_{\text{latent}} = 128$, and $C_{\text{latent}} = C^{\text{dir}}_{\text{latent}} + C^{\text{dist}}_{\text{latent}} = 256$.

B.2. Architecture details of 3D gaze transformer

The architecture is illustrated in Fig. 1. We use the transformer module [25] from PyTorch [17]. The object encoder f_{encoder} and the gaze decoder f_{decoder} consist of $N_{\text{encoder}} = 3$ transformer encoder layers and $N_{\text{decoder}} = 3$ transformer decoder layers, respectively. The feedforward network has a dimension of 512, and the multi-head attention [25] uses 2 heads. Other hyperparameters follow the default settings in [17]. The transformer processes the object feature E_{object} as the source sequence and the subject feature E_{subject} as the target sequence.

Following [24], we incorporate a gaze-cone-based additive attention bias $B \in \mathbb{R}^{N_{\text{object}}}$ to the object features F_{object} in the cross-attention between E_{subject} and F_{object}. This bias emphasizes object features aligned with the subject's view direction $\mathbf{v}$, where each element of the bias is defined as the cosine similarity between $\mathbf{v}$ and $\mathbf{p}_{\text{object}} \in P_{\text{object}}$.

For batch processing in the attention layers, the number of object positions is padded to a maximum $N^{\text{max}}_{\text{object}} \geq N_{\text{object}}$.

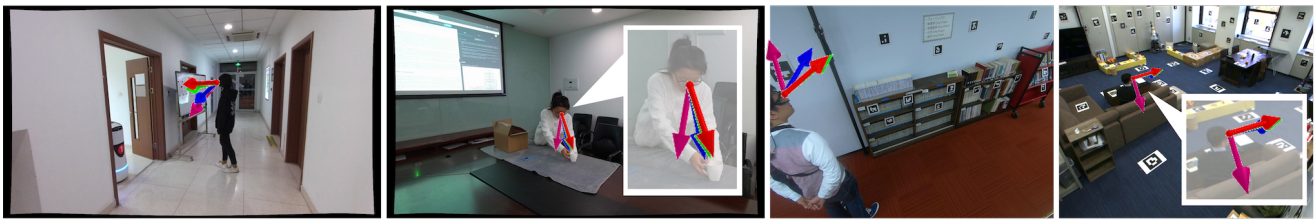

Figure 6. Qualitative ablation results illustrating the effects of pose and object understanding. The red arrow indicates the ground truth, while the magenta, blue, and green arrows represent predictions from the *Appearance*, *Appearance + Pose*, and *Appearance + Pose + Object* models, respectively, as described in Tab. 4 of the main paper. (a) and (b) present results from the GFIE dataset [6], while (c) and (d) show results from the GAFA dataset [14].

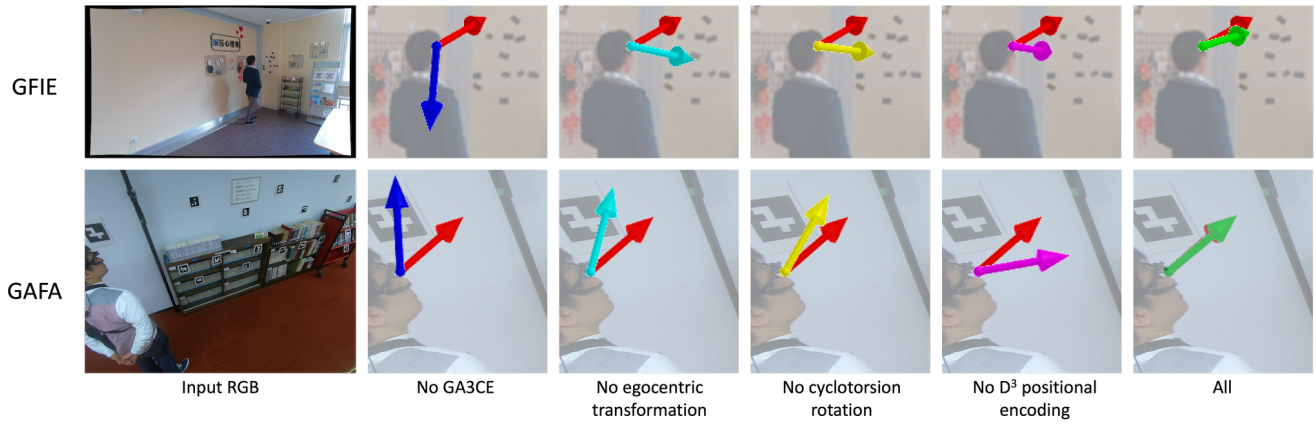

Figure 7. Qualitative ablation results for gaze-aware 3D context encoding and the proposed components. The top row shows results on the GFIE [6] dataset, while the bottom row presents results on the GAFA [14] dataset. The red arrow indicates the ground truth, and the other arrows represent outputs from ablated models. *All* denotes the full model with all proposed components.

Specifically, $N_{\text{object}}^{\text{max}}$ is set to 168 for the GFIE [6] and CAD-120 [8] datasets, and 278 for the GAFA dataset [14]. During multi-head attention, non-existent object positions are masked out.

Finally, the residual gaze direction $\mathbf{g}'$ is decoded using a two-layer MLP with 512 hidden units and ReLU activation.

B.3. Training details

The batch size is set to 32 for the GFIE dataset [6] and 64 for the GAFA dataset [14] to accommodate the larger dataset size and improve training efficiency. The network is trained for 20 epochs on a single A10 GPU using the AdamW optimizer [12] with a learning rate of 0.0014. Cosine scheduling with a 4-epoch warm-up is employed. Weight decay is set to 0.1, and gradient clipping with an L2-norm threshold of 0.1 is applied.

For the GFIE dataset, noise ϵ is added to the view direction $\mathbf{v}$ as an augmentation to mitigate overfitting. The perturbed view direction $\mathbf{v}_{\text{noise}}$ is computed as $\mathbf{v}_{\text{noise}} = \frac{\mathbf{v}+\epsilon}{\|\mathbf{v}+\epsilon\|}$, where $\epsilon \in [-0.5, 0.5]^3$. This results in an average angular shift of approximately 22 degrees. For the GAFA dataset, however, this augmentation increased validation error and was therefore only applied during training on the GFIE dataset.

B.4. Architecture details of the appearance-based estimators

The appearance-based estimator in the baseline GFIE [6] uses a ResNet50 image encoder followed by a gaze prediction head composed of MLPs. It takes an RGB head image as input and outputs a 3D gaze direction represented as a unit vector.

In the baseline GAFA [14], the appearance-based estimator corresponds to the *Head and Body Network* described in [14]. It processes seven temporal frames: the target frame, along with three future and three past frames. Each frame includes a full-body RGB image, a 2D head position mask, and the 2D velocity of the body center. The full-body image and head position mask are encoded separately using 2D convolutional networks, while the body velocity is en-

Figure 8. Visualization of cases where objects are missing along the gaze direction. Red and green arrows indicate the ground truth and the prediction, respectively. In (a), objects lie outside the gaze direction. In (b), no objects are present near the gaze direction.

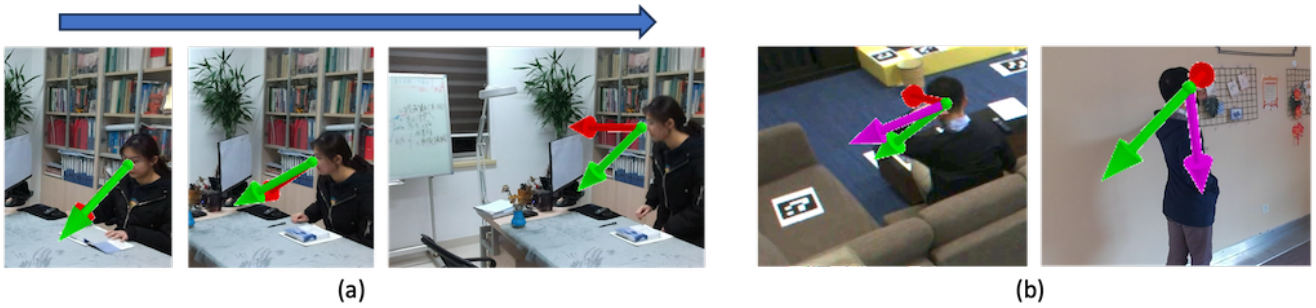

Figure 9. Visualization of failure cases. Red and green arrows denote the ground truth and the predictions, respectively. In (a), the blue arrow indicates the temporal direction. In (b), the magenta arrow represents the view direction $\mathbf{v}$.

coded using MLPs. These three features are concatenated and passed through LSTM layers, which predict the directions and uncertainties of both the body and head for each frame, modeled as parameters of a 3D von Mises-Fisher distribution.

B.5. Gaze-following modules (GFM) [6]

GFM, as referenced in Tabs. 1 and 2, refers to the 2D/3D gaze-following modules from [6]. The 2D module is a ResNet50-based convolutional autoencoder that takes as input a multi-channel 2D feature comprising a scene RGB image, a head position mask, and a field-of-view (FoV) feature map. It outputs a 2D heatmap indicating the gazed point. The FoV feature map encodes the pixel-wise directional similarity between the estimated gaze direction (obtained from the head image, as described in Supp. B.4) and each 3D point backprojected from the depth map using the provided camera intrinsics. All 3D points are normalized relative to the known 3D head position.

The 3D gaze-following module takes the 2D heatmap and estimated gaze direction as input and outputs the 3D gazed point. This module is non-learnable and deterministic: it selects the backprojected 3D point corresponding to the pixel location with the highest similarity in the FoV feature map near the peak of the 2D heatmap. The final 3D gaze direction is then computed as the vector from the known head position to the estimated 3D point.

B.6. Baseline modification for 3D context input

As described in Sec. 4, we modified the baseline methods GFIE [6] and GAFA [14] to incorporate 3D pose P_{pose} and object positions P_{object}, aligning their inputs with our approach for the corresponding datasets. Specifically, the pose P_{pose} is normalized using the head position $\mathbf{t}_{\text{head}}$ and head size s, resulting in $\hat{P}_{\text{pose}}$. Similarly, object positions P_{object} are normalized by the head position $\mathbf{t}_{\text{object}}$ and scaled so that their largest extent equals one, yielding $\hat{P}_{\text{object}}$.

These modifications are illustrated in Fig. 2, which highlights the updated modules in the pipeline while omitting other baseline components for clarity. Full pipeline details are available in [6] and [14]. As noted in Sec. 4, we refer to the modified methods as *GFIE + 3D* and *GAFA + 3D*.

GFIE + 3D was trained using the publicly available code from [6]. In the case of GAFA [14], training the entire pipeline, including Head and Body Network along with the gaze estimation module, as shown in Fig. 2 (b), led to overfitting without convergence. Improved results were obtained by training only the gaze estimation module while keeping the pre-trained weights for Head and Body Network frozen. For all other training settings, refer to [14].

B.7. Pipeline modification for ablation study on pose and object

For the *Appearance* model in Tab. 4 of the main paper, we use the same pre-trained model with f_{view} as the appearance-based gaze direction estimator from the head image for the GFIE dataset [6]. For the GAFA dataset [14], we use the pre-trained gaze direction module from the GAFA baseline, which also takes the head direction $\mathbf{v}$ from the same model with f_{view}. Since these appearance-only models lack 3D context input, GA3CE is not applied. For the *Appearance + Pose* model, which does not use object input, a constant latent vector is used as F_{object} in the input to the decoder f_{decoder} to exclude object information.

B.8. Depth map processing

We utilize the zero-shot metric depth estimator [1] for the RGB-only experiments on the GFIE dataset [6] in Sec. 4.1, as well as for all evaluations on the GAFA dataset [14] in Sec. 4.3.

For backprojecting object positions, as discussed in Sec. 3.1, we complete missing regions in the depth maps of the GFIE [6] and the CAD-120 [8] datasets using depth completion [11].

C. Additional results

C.1. Additional results on the GFIE dataset [6]

Additional qualitative results Additional qualitative results on the GFIE dataset [6] are shown in Fig. 3.

Comparison to the baseline GFIE [6] + the SOTA 2D gaze-following method We further compare our proposed approach with the baseline GFIE [6], replacing its 2D gaze-following module with the latest state-of-the-art method, ViTGaze [21]. ViTGaze utilizes the large-scale pre-trained model DINOv2 [15] as its backbone image encoder. DINOv2 has demonstrated strong capabilities in both object- and scene-level understanding, implicitly learning part-level instance features for diverse categories, including objects and body parts. It also provides a foundation for 3D spatial understanding, such as depth estimation.

It is important to emphasize that the focus of this paper is not to improve or compete with 2D gaze-following methods. As discussed in Sec. 1, these methods assume different task settings, such as requiring the gaze target to be visible when providing gaze information. In contrast, 3D gaze direction estimation can provide gaze information even when the target is not visible.

We first trained the ViTGaze model and replaced GFIE's 2D gaze-following module during inference. Quantitative evaluation results are shown in Tab. 1. Incorporating ViTGaze improves 2D gaze-following performance, reducing the L2 distance between ground truth and gaze points by 17%. Nevertheless, our method still achieves superior results in 3D gaze direction estimation. This is likely because even small errors in the detected gaze point on the image can lead to larger errors in 3D space, especially in the presence of significant depth variation across pixels.

C.2. Additional results on the CAD-120 dataset [8]

Full quantitative results Quantitative results on the CAD-120 dataset [8], including comparisons with other baselines and GFIE [6], are shown in Tab. 2. The proposed method outperforms the baseline methods.

Additional qualitative results Additional qualitative results for the CAD-120 dataset [8] are provided in Fig. 4.

C.3. Additional qualitative results on the GAFA dataset [14]

Additional qualitative results for the GAFA dataset [14] are shown in Fig. 5.

C.4. Qualitative results for ablation studies

3D Understanding of Pose and Object Fig. 6 shows qualitative results for 3D understanding of pose and objects. Blue arrows indicate that combining appearance and pose improves results compared to using appearance alone, marked by magenta arrows. In examples (c) and (d), this combination effectively corrects incorrect estimations, leading to more accurate predictions. Adding object information, represented by green arrows in the *Appearance + Pose + Object* setting, further refines the results, aligning the estimated directions more closely with the ground truth.

Gaze-aware 3D context encoding (GA3CE) Fig. 7 presents qualitative results for GA3CE. Disabling GA3CE (*No GA3CE*) results in less accurate predictions, as indicated by the blue arrows across both datasets [6, 14]. Incorporating all proposed components (*ALL*) yields the most accurate results, highlighted by the green arrows.

C.5. Robustness to view direction

To assess robustness to the view direction $\mathbf{v}$ as a directional prior, we add noise to the ground truth direction so that the resulting $\mathbf{v}$ has a 3D MAE matching a target value. When the 3D MAE of $\mathbf{v}$ is 20.0, 30.0, and 40.0 on the GFIE dataset [6], the model's corresponding 3D MAEs are 11.11, 13.38, and 16.69. On the GAFA dataset [14], 3D MAEs of 20.0, 29.7, and 38.6 for $\mathbf{v}$ result in corresponding errors of 18.45, 24.67, and 30.95. The proposed method demonstrates greater robustness on the GFIE dataset [6], likely due to more frequent close object interactions that aid accurate gaze prediction.

C.6. Robustness to object absence

We examine how the model performs when objects are either outside the subject's gaze direction or largely absent from the scene. Objects are removed from the images using the inpainting tool [18], with results shown in Fig. 8. In (a), the 3D MAE is 14.37 (view direction $\mathbf{v}$: 15.38), and in (b), the 3D MAE is 10.28 (view direction $\mathbf{v}$: 27.22). In both cases, the model estimates reasonable gaze directions by leveraging pose and view direction as context cues, even when objects are present but not gazed, or are absent.

D. Failure cases

Typical failure cases are illustrated in Fig. 9. In (a), the method fails to track gaze shifting from right to left. Since it does not incorporate temporal information, it struggles in situations where pose and object cues alone are insufficient to resolve directional ambiguity. In (b), the model fails to predict a reasonable gaze direction when the view direction $\mathbf{v}$ contains significant error. As the method relies on $\mathbf{v}$ as a directional prior in the egocentric frame, it cannot recover from a highly inaccurate view direction.

E. Limitation and future work

While our method demonstrates strong performance within the trained domains, as shown in Tabs. 1 and 3, its generalization to unseen domains leaves room for improvement. This is evident in the results on the CAD-120 dataset Tab. 2, where Gaze360 [7], despite relying solely on head appearance, outperforms Ours without GFM. Although our method effectively leverages the view direction $\mathbf{v}$ as a prior, it inherits limitations when this prior is suboptimal as discussed in Supp. D, particularly in unseen domains. We attribute the suboptimal performance on the CAD-120 dataset to this issue, as seen in the $\text{GFIE}_{\text{head}}$ results in Tab. 2 where the 3D MAE is significantly higher than that of Gaze360 (Gaze360: 20.6, $\text{GFIE}_{\text{head}}$: 27.3). When the view direction is adjusted to match the quality of Gaze360 (view direction 3D MAE: 20.2), following a similar procedure to the experiments in Supp. C.5, our method achieves a 3D MAE of 18.7, outperforming Gaze360. This suggests that our method is more sensitive to the quality of the view direction in unseen domains compared to its robustness in trained domains, as shown in Supp. C.5. As illustrated by the Ours + GFM results in Tab. 2, incorporating GFM significantly improves performance, even with suboptimal view direction estimates. This highlights a promising direction for improving generalization by integrating gaze-following approaches.

Our method currently assumes the subject's head is visible to the camera without occlusion to enable 3D localization using backprojection with a depth map and head bounding box. Extending this approach to a multi-view setting could address this limitation, and we plan to explore this in future work.

Additionally, our pipeline does not yet incorporate object semantics, which has been shown to be a promising approach for considering scene context in 2D gaze following [22]. Expanding our method to include semantic cues alongside object locations is an interesting future direction.

Finally, the current pipeline focuses on a single subject per scene, consistent with the previous works [6, 14]. Future research will explore multi-person scenarios to capture spatial relationships, enabling tasks such as 3D joint attention estimation.

References

[1] Bhat, S.F., Birkl, R., Wofk, D., Wonka, P., Müller, M.: Zoedepth: Zero-shot transfer by combining relative and metric depth. arXiv preprint arXiv:2302.12288 (2023) 2, 6

[2] Chong, E., Wang, Y., Ruiz, N., Rehg, J.M.: Detecting attended visual targets in video. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2020) 2

[3] Einhäuser, W., Spain, M., Perona, P.: Objects predict fixations better than early saliency. Journal of vision **8**(14) (2008) 1

[4] Findlay, J.M., Gilchrist, I.D.: Active vision: The psychology of looking and seeing. No. 37, Oxford University Press (2003) 1

[5] Fischer, T., Chang, H.J., Demiris, Y.: Rt-gene: Real-time eye gaze estimation in natural environments. In: Proceedings of the European conference on computer vision (2018) 2

[6] Hu, Z., Yang, Y., Zhai, X., Yang, D., Zhou, B., Liu, J.: Gfie: A dataset and baseline for gaze-following from 2d to 3d in indoor environments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023) 1, 2, 3, 4, 5, 6, 7

[7] Kellnhofer, P., Recasens, A., Stent, S., Matusik, W., Torralba, A.: Gaze360: Physically unconstrained gaze estimation in the wild. In: Proceedings of the IEEE/CVF international conference on computer vision (2019) 2, 7

[8] Koppula, H.S., Gupta, R., Saxena, A.: Learning human activities and object affordances from rgb-d videos. The International journal of robotics research **32**(8) (2013) 2, 3, 4, 6

[9] Lee, J.K., Yoon, K.J.: Real-time joint estimation of camera orientation and vanishing points. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1866–1874 (2015) 1

[10] Lian, D., Yu, Z., Gao, S.: Believe it or not, we know what you are looking at! In: Proceedings of the Asian Conference on Computer Vision (2018) 2

[11] Liu, T.Y., Agrawal, P., Chen, A., Hong, B.W., Wong, A.: Monitored distillation for positive congruent depth completion. In: Proceedings of the European Conference on Computer Vision (2022) 6

[12] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) 4

[13] Ma, C.Y., Hang, H.M.: Learning-based saliency model with depth information. Journal of vision **15**(6) (2015) 1

[14] Nonaka, S., Nobuhara, S., Nishino, K.: Dynamic 3d gaze from afar: Deep gaze estimation from temporal eye-head-body coordination. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022) 2, 3, 4, 5, 6, 7

[15] Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.Y., Xu, H., Sharma, V., Li, S.W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision (2023) 6

[16] Park, H.S.: Where am i? using vanishing points. https://www-users.cse.umn.edu/~hspark/CSci5980/Lec4_Localization(VanishingPoint).pdf, lecture notes for CSci 5980: Computer Vision, University of Minnesota 1

[17] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019) 3

[18] PhotoRoom: Remove object from photo - PhotoRoom (2024), https://www.photoroom.com/tools/remove-object-from-photo, accessed: 2025-03-24 7

[19] Recasens, A., Khosla, A., Vondrick, C., Torralba, A.: Where are they looking? In: Advances in neural information processing systems (2015) 2

[20] Sárándi, I., Hermans, A., Leibe, B.: Learning 3D human pose estimation from dozens of datasets using a geometry-aware autoencoder to bridge between skeleton formats. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (2023) 1

[21] Song, Y., Wang, X., Yao, J., Liu, W., Zhang, J., Xu, X.: Vitgaze: gaze following with interaction features in vision transformers. Vision Intelligence **2**, 31 (2024) 1, 6

[22] Tafasca, S., Gupta, A., Bros, V., Odobez, J.m.: Toward semantic gaze target detection. In: Advances in neural information processing systems (2024) 7

[23] Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. Advances in neural information processing systems **33** (2020) 3

[24] Tonini, F., Dall'Asen, N., Beyan, C., Ricci, E.: Object-aware gaze target detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2023) 3

[25] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems (2017) 3