De^{2}Gaze: Deformable and Decoupled Representation Learning for 3D Gaze Estimation

Supplementary Material

Appendices

Section 1 describes the implementation process of the method in more details. We have carried out additional experiments, and the results will be explained in Section 2. Finally, in Section 3, we will discuss the limitations of the method and future work. Our code will be released on GitHub upon acceptance.

1. More Details

1.1. Render Semantics

We begin by generating the template point clouds for both the pupil and the iris using polar coordinates. The process involves discretizing the angles and radii, and then converting the polar coordinates into 3D Cartesian coordinates.

Generate Angle Grid. We generate a set of angles in the range $[0, 2\pi)$ for each batch and frame. This is done by discretizing the angle space into N_{angles} points:

$$\theta_i = \frac{2\pi i}{N_{\text{angles}}}, \quad \text{for} \quad i = 0, 1, 2, \ldots, N_{\text{angles}} - 1, \tag{1}$$

this results in an angle grid of shape $[B, N_{\text{angles}}]$, where B is the batch size multiplied by the number of frames.

Generate Radius Grids. The radius of the pupil and iris are discretized into N_{radius} points. For the pupil, the radii range from 0 to r_{pupil}, while for the iris, they range from r_{pupil} to r_{iris}. The radii are generated as:

$$\begin{aligned} r^i_{\text{pupil}} &= \frac{i}{N_{\text{radius}}} r_{\text{pupil}}, \\ r^i_{\text{iris}} &= \frac{i}{N_{\text{radius}}} (r_{\text{iris}} - r_{\text{pupil}}) + r_{\text{pupil}}, \\ &\text{for} \quad i = 0, 1, 2, \ldots, N_{\text{radius}} - 1, \end{aligned} \tag{2}$$

this generates two radius grids, one for the pupil and one for the iris, both of shape $[B, N_{\text{radius}}]$.

Convert Polar Coordinates to Cartesian Coordinates. We convert the polar coordinates into 3D Cartesian coordinates for both the pupil and the iris. For each pair of radius r and angle θ, the Cartesian coordinates (x, y, z) are computed as:

$$x = r \cdot \cos(\theta), \quad y = r \cdot \sin(\theta). \tag{3}$$

For the pupil and iris point clouds, the z-coordinate is set to a fixed value, determined by the distance from the camera, L_p, and inverted to place the point clouds in front of the camera:

$$z_{\text{pupil}} = z_{\text{iris}} = -L_p \tag{4}$$

Thus, the final 3D coordinates for the pupil and iris are:

$$\begin{aligned} \mathbf{P}_{\text{pupil}} &= (x_{\text{pupil}}, y_{\text{pupil}}, z_{\text{pupil}}), \\ \mathbf{P}_{\text{iris}} &= (x_{\text{iris}}, y_{\text{iris}}, z_{\text{iris}}). \end{aligned} \tag{5}$$

The resulting point clouds have shapes $[B, N_{\text{angles}} \times N_{\text{radius}}, 3]$.

1.2. Regression Head

Following previous studies [1–4], we apply a linear projection to the variable-level content vector Q^{con}_{var} and invariant-level content vector Q^{con}_{inv} to generate the eye parameters E_{var} and E_{inv}:

$$\begin{aligned} E_{var} &= \text{Linear}(Q^{con}_{var}), \\ E_{inv} &= \text{Linear}(Q^{con}_{inv}). \end{aligned} \tag{6}$$

The generated variable and invariant parameters jointly reconstruct a three-dimensional eyeball model, and project it on a two-dimensional plane to generate the edge contours of the pupil and iris as the initial sampling positions. Moreover, we employ a Multi-Layer Perception (MLP) with ReLU activation to generate sampling offsets. Specifically, the invariant-level content vectors Q^{con}_{inv} are used to compute invariant-level offsets, while the variable-level content vectors Q^{con}_{var} are utilized to generate variable-level offsets:

$$\begin{aligned} \Delta^{inv}_{hn} &= \text{MLP}(Q^{con}_{inv}), \\ \Delta^{var}_{hn} &= \text{MLP}(Q^{con}_{var}). \end{aligned} \tag{7}$$

These offsets are subsequently employed to refine position vectors Q^{pos}_{inv} and Q^{pos}_{var}. The regression head is appended to the final encoder layer as well as each decoder layer. Furthermore, to preserve the alignment between dual-level queries, we share the ground truth of pupil and iris edge among each aligned query.

1.3. Training Details

De^{2}Gaze is trained on NVIDIA A100 GPU, with a batch size of 128. We set the initial weights of projection edge loss λ_{edge}, eyeball center loss λ_{eyeC} and pupil center loss λ_{pupilC} to 0.15. The initial weight of that two 3D gaze loss λ^{L2}_{gaze} and $\lambda^{cos-sin}_{gaze}$ are set to 2.5.

Subject	Loss	TEyeD-subset_A				
		3D gaze [°]↓	2D gaze [°]↓	Sem. Iou	2D pupil cent.[px]↓	2D eye cent.[px]↓
subject1	Gaze	0.49	2.10	N/A	N/A	N/A
	Sem. + Gaze + Cent.	0.36	1.77	87.5%	3.62	9.63
subject2	Gaze	0.82	3.50	N/A	N/A	N/A
	Sem. + Gaze + Cent.	0.59	2.22	88.3%	2.45	1.70
subject3	Gaze	0.50	2.05	N/A	N/A	N/A
	Sem. + Gaze + Cent.	0.48	1.88	90.4%	0.82	0.92
subject4	Gaze	0.56	2.18	N/A	N/A	N/A
	Sem. + Gaze + Cent.	0.49	1.88	83.5%	3.51	12.78
subject5	Gaze	0.74	4.74	N/A	N/A	N/A
	Sem. + Gaze + Cent.	0.62	4.57	86.4%	3.71	5.40

Table 1. Quantitative results of individual training and testing of five subjects on TEyeD dataset. After eliminating the influence of kappa angle, the results show that applying more constraints is beneficial to improve the reconstruction accuracy of 3D eyeball model.

2. Additional Experiments

Impact of the number of reference points. We varied the number of sampling points and evaluated their impact on both model performance and computational cost. The results in Fig. 1 demonstrate that:

With a smaller number of sampling points, the attention mechanism struggles to capture sufficient spatial details, leading to degraded performance, especially in scenarios with complex or high-resolution inputs. While computational efficiency is significantly improved, the loss of important features limits the overall accuracy of the model.

Increasing the number of sampling points initially enhances the model's ability to capture fine-grained details. However, when the number of points becomes too large, performance starts to degrade. This is because an excessive number of sampling points introduces noise and irrelevant features, which interfere with the attention mechanism, leading to reduced precision. Additionally, the increased computational cost and memory usage further impact the overall efficiency.

The position visualization of different numbers of sampling points is shown in Fig. 2. By carefully selecting an optimal number of sampling points, the model achieves a balance between computational efficiency and accuracy. This configuration allows for robust feature extraction while minimizing the influence of noise and redundant information.

Impact of the kappa angle offset between the optical and the visual axes. The normalized optical axis g is defined as the vector from the eyeball center o_e to the iris center o_i, $g = \frac{o_i - o_e}{\|o_i - o_e\|}$. We consider g the approximated gaze vector. Note that we do not model the kappa angle offset between the optical and the visual axes. In our previous experiments,

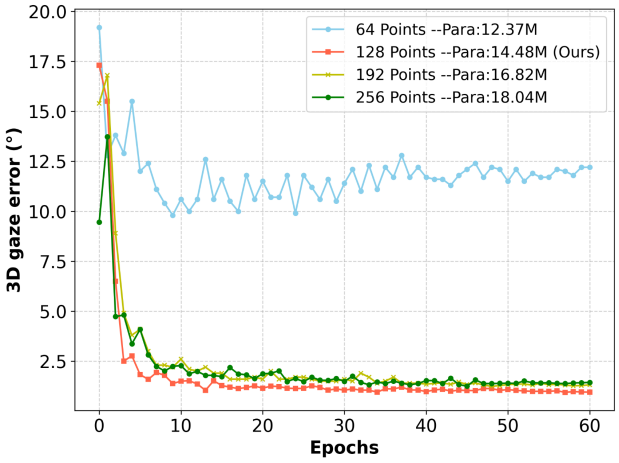

Figure 1. Compare the 3D gaze error curves under four different sampling numbers during training to choose the best sampling number.

we put more supervision on the whole eyeball, such as the center of the eyeball and pupil, as well as the edge of the projection. However, the accuracy has declined. We trained and tested five subjects separately to eliminate the influence of kappa angle difference among different subjects. The experimental results in Tab. 1 show that imposing more constraints on the same subject can finally improve the accuracy of eyeball fitting and 3D gaze estimation.

3. Limitation and Future Work

3.1. Limitations

While De2Gaze achieves state-of-the-art results in 3D gaze estimation, several limitations remain:

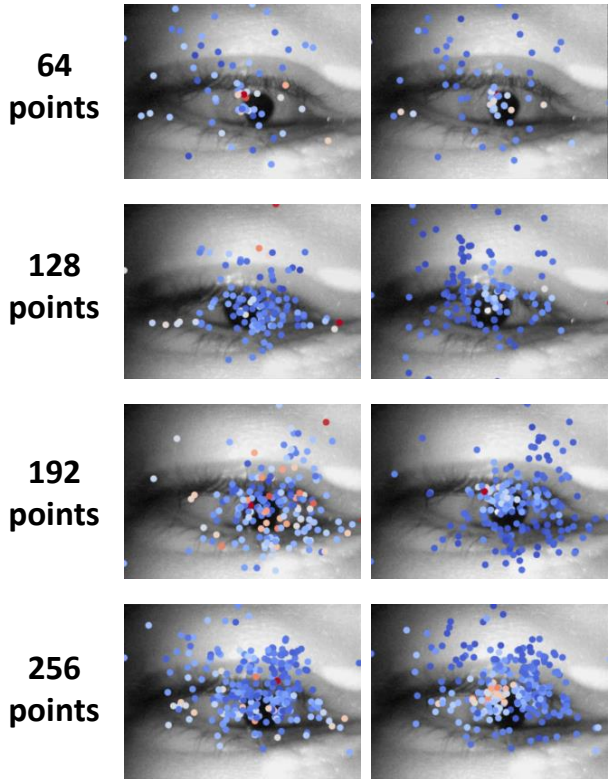

Figure 2. Visualization of different number of sampling points. When the number of sampling points is 128, the region of interest is more concentrated in the eyelid and is least disturbed by the characteristics of irrelevant regions.

Dependency on 3D Eyeball Model Precision. The accuracy of the learnable 3D eyeball model plays a critical role in the success of De^{2}Gaze. Errors in estimating invariant parameters (e.g., eyeball radius, center position) can propagate to gaze direction predictions, especially under non-ideal lighting conditions or when partial occlusions occur.

Limited Temporal Context Modeling. Although De^{2}Gaze processes sequential frames, the method does not fully leverage long-term temporal dependencies. As a result, it may struggle with tasks requiring understanding of extended eye movement patterns, such as saccades or fixations over prolonged periods.

Projection-Based Sampling Bias. The deformable sparse attention mechanism relies on projecting 3D geometric features onto the 2D plane for sampling. This design assumes accurate alignment between 3D reconstructions and image semantics, which may not hold in cases with significant calibration errors or out-of-distribution (OOD) inputs.

3.2. Future Work

To address these limitations and further advance the proposed De^{2}Gaze framework, future work will focus on the following aspects:

Enhancing 3D Eyeball Model Robustness. Incorporate additional constraints, such as anatomical priors or multi-view data, to improve the robustness of invariant parameter predictions, particularly under challenging conditions like occlusions or extreme lighting.

Exploiting Long-Term Temporal Dependencies. Introduce recurrent architectures or temporal transformers to capture extended eye movement patterns. This would enable De^{2}Gaze to perform well in tasks requiring dynamic gaze analysis over longer sequences.

Reducing Projection-Based Sampling Errors. Investigate adaptive refinement techniques for 3D-to-2D projection points to mitigate the impact of misaligned geometric features, and explore hybrid sampling strategies combining dense and sparse representations.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1

[2] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13526–13535, 2021.

[3] Jing Tan, Xiaotong Zhao, Xintian Shi, Bin Kang, and Limin Wang. Pointtad: Multi-label temporal action detection with learnable query points. *Advances in Neural Information Processing Systems*, 35:15268–15280, 2022.

[4] Guozhen Zhang, Yuhan Zhu, Haonan Wang, Youxin Chen, Gangshan Wu, and Limin Wang. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5682–5692, 2023. 1