# GazePyFormer: Capturing Multi-Scale Spatial Contexts for Gaze Following

Haiying Xia[1], Ruihan Yang[1], Yumei Tan[1], Shuxiang Song[1*]
[1] Guangxi Key Laboratory of Brain-inspired Computing and Intelligent Chips,
School of Electronic and Information Engineering, Guangxi Normal University,
Guilin, China
{xhy22, 2629478004, tanyumei, songshuxiang}@gxnu.edu.cn

***Abstract*—Gaze following aims to predict the target location of a person's gaze by integrating scene context with head cues. While multi-scale representations are essential for handling targets of varying sizes, existing Transformer-based methods often rely on fixed-resolution feature maps or aggregate features in a post-hoc manner. Such designs fail to explicitly capture spatial dependencies across different scales, leading to sub-optimal localization in complex scenes where targets may be distant or small. To address this, we propose GazePyFormer, a Multi-scale Pyramid Pooling Transformer that performs cross-scale reasoning within the encoding process. Specifically, we develop a GazePyFormer Encoderthat embeds hierarchical pyramid pooling into the Transformer blocks to extract features across diverse receptive fields. This is followed by a GazeModulator, which uses Spatial-FiLM to adaptively align gaze priors with multi-level scene features. Finally, a Multi-scale Fusionmodule integrates these coarse-to-fine representations to produce precise gaze heatmaps. Experiments on the GazeFollow and VideoAttention-Target benchmarks demonstrate that GazePyFormer consistently outperforms state-of-the-art methods, showing superior robustness and accuracy in challenging environments.**

***Index Terms*—Gaze following, Multi-scale feature, Pyramid pooling, Gaze-guided fusion**

## I. Introduction

Gaze is an important cue for human perception, revealing attention and intention while facilitating environmental understanding. Gaze following has broad applications in cognitive science [1], human-computer interaction [2], and neuroscience [3]. Among related tasks, gaze following predicts an individual's gaze target in a single image, particularly in complex multi-person scenes, and has attracted increasing attention in computer vision.

Deep learning has substantially advanced gaze following research, with representative approaches based on Convolutional Neural Networks (CNNs) [4]–[13] and Graph Neural Networks (GNNs) [14], [15]. Early CNN based methods typically adopted dual stream architectures, in which head cues and scene context were processed separately and then fused to predict gaze targets. While effective at capturing local patterns, these models struggled to model long range dependencies between the head and the scene. GNN based approaches further modeled interactions between people and objects but often relied on explicit object detection or handcrafted graph structures, limiting flexibility.

* Corresponding author.

More recently, Transformer-based architectures have become a mainstream solution for gaze following because of their strong global reasoning ability. Prior works [16]–[20] model person-scene interactions with cues such as head position, gaze direction, and auxiliary tokens. MTGS [21] introduces semantic cues from vision-language models through early fusion, Sharingan [22] uses a Conditional DPT decoder for multi-scale feature fusion, and SocialFusion [23] incorporates structured object-level information via lightweight vision-language fusion. However, most existing methods still rely on fixed-resolution features or post-hoc multi-scale aggregation, which limits continuous spatial hierarchy modeling.

To address these limitations, we propose GazePyFormer, a novel hierarchical framework that captures and integrates multi-scale spatial contexts via an end-to-end pyramid structure. Its core lies in sustaining a continuous spatial hierarchy throughout encoding and fusion. Specifically, the GazePyFormer Encoder incorporates hierarchical pyramid pooling within Transformer blocks to explicitly extract multi-receptive-field features, enabling simultaneous capture of local texture and global semantics in each layer—distinct from standard Transformers. To bridge person-specific orientation and scene-level content, we develop a GazeModulator, which leverages Spatial-FiLM and adaptive pooling to dynamically modulate hierarchical scene features with gaze priors, ensuring gaze awareness across all spatial scales. A Progressive Multi-scale Fusion module further integrates these coarse-to-fine representations to generate robust, high-resolution gaze heatmaps. Extensive experiments on multiple benchmarks demonstrate consistent substantial performance gains, validating GazePyFormer's effectiveness and robustness in complex scenarios.

In summary, our main contributions are as follows:

- We introduce a GazePyFormer Encoder, which incorporates a multi-scale pooling mechanism directly into the Transformer stages, enabling the extraction of rich, scale-invariant spatial representations.
- We propose a GazeModulator, a novel component that aligns head-pose priors with multi-level scene features through spatial modulation, effectively reducing localization ambiguity in complex visual fields.
- Extensive experiments conducted on the public benchmark datasets GazeFollow and VideoAttentionTarget further demonstrate the superior performance of GazePyFormer.

## II. METHODOLOGY

### A. Overview of the GazePyFormer

As illustrated in Fig. 1, GazePyFormer consists of three main modules: the GazePyFormer Encoder, GazeModulator, and multi-scale fusion module. As shown in the dashed box, the encoder comprises four hierarchical stages. The input image is first partitioned into patches, which are embedded and augmented with learnable positional encodings. At each stage, neighboring patches are progressively aggregated into higher dimensional representations, producing multi-scale scene features $f^i_{scene}$, where $i \in 1, 2, 3, 4$ denotes the stage index. In parallel, the cropped head image and head bounding box are encoded into a gaze vector $V_{gaze}$, which captures person-specific appearance and spatial cues. Conditioned on $V_{gaze}$, the GazeModulator adaptively modulates the hierarchical scene features to align gaze priors with scene representations. The modulated features are then fused across stages to combine global context with local details for gaze heatmap prediction. In addition, the fourth-stage features are used for in/out-of-frame classification.

### B. GazePyFormer Encoder

Gaze following requires both fine-grained cue modeling and global scene understanding, as gaze targets may appear near the head or far across the scene. Conventional ViTs, however, rely on single-scale representations and thus have limited ability to capture such hierarchical relations. To address this, we propose the GazePyFormer Encoder, inspired by P2T [24], which integrates pyramid pooling and hierarchical Transformer encoding for robust multi-scale gaze following.

Fig. 2(a) shows a single GazePyFormer Encoder block. It first reshapes patch embeddings into a two-dimensional feature map to restore spatial structure, and then applies pyramid pooling-based multi-head self-attention (P-MHSA) to capture both local head cues and global scene context. The representation is further refined by residual connection, Layer Normalization (LN), and an inverted residual bottleneck (IRB) [25] as the feed-forward network (FFN). The process is formulated as follows:

$$X_{attn} = LN\left(X + P - MHSA\left(X\right)\right), \tag{1}$$

$$X_{out} = LN\left(X_{attn} + FFN\left(X_{attn}\right)\right), \tag{2}$$

where $LN\left(\right)$ denotes the LayerNorm operation, and $FFN\left(\right)$ denotes the IRB module. $X \in \mathbb{R}^{C\times H\times W}$ denotes the reshaped input feature map. $X_{\text{attn}} \in \mathbb{R}^{C\times N}$ (where $N = H \times W$) denotes the output of the P-MHSA module in a flattened token sequence, while $X_{\text{out}} \in \mathbb{R}^{C\times H\times W}$ is the final output of the encoder block.

The P-MHSA module is shown in Fig. 2(b). It applies four average pooling operations with different ratios to $X$ to generate multi-scale feature maps. This design preserves local head details while suppressing background noise and redundancy, thereby enriching the diversity of keys and queries in attention. As a result, the model can jointly attend to the head and surrounding scene context. The pyramid pooling operations are formulated as follows:

$$P_i = AvgPool_i\left(X, r_i\right), \tag{3}$$

where $P_i$ denotes the generated pyramidal feature maps at different pooling ratios $r_i$ (for $i = 1, 2, 3, 4$). These pooled features are flattened and concatenated along the channel dimension, and the result is normalized to ensure stable feature distributions and facilitate effective multi-scale feature fusion. The operation is as follows:

$$Y = LN\left(flat\&cat(P_1, P_2, P_3, P_4)\right), \tag{4}$$

where $Y \in \mathbb{R}^{C\times M}$, with $M$ being the total number of pooled tokens. Through this process, multi-scale pooling integrates contextual information from different receptive fields, allowing $Y$ to capture both local details and global semantics.

Unlike the standard ViT, our approach derives the queries ($Q$) directly from the flattened input $X$, while the keys ($K$) and values ($V$) are linearly projected from $Y$. As $Y$ encapsulates an abstract contextual representation of $X$, it acts as an effective surrogate for $X$ within the multi-head self-attention mechanism, enabling richer contextual interactions and more robust feature learning. The computation is as follows:

$$(Q, K, V) = (XW_q, YW_k, YW_v), \tag{5}$$

where $W_q$, $W_k$, and $W_v$ are the corresponding learnable weights; $Q \in \mathbb{R}^{C\times N}$ and $K, V \in \mathbb{R}^{C\times M}$. Subsequently, $Q$, $K$ and $V$ are fed into the multi-head self-attention mechanism and computed as follows:

$$A = Soft\max\left(\frac{QK^T}{\sqrt{d_K}}\right)V, \tag{6}$$

where $d_K$ denotes the channel dimension of $K$. $A \in \mathbb{R}^{C\times N}$ represents the attention features. Since $K$ and $V$ are derived from pooled features, their sequence lengths $M$ typically much shorter than that of the original input $X$ (i.e., $M \ll N$), which substantially improves computational efficiency.

### C. GazeModulator

Integrating gaze priors with scene features is essential for precise gaze target localization. Existing fusion strategies, either early concatenation or mid-level feature fusion, often lose semantic information or are computationally inefficient for 2D feature maps. To address this, we propose a GazeModulator, which directly fuses gaze priors with multi-scale scene features on 2D encoder feature maps, as illustrated in Fig. 3.

The cropped head image is processed by a gaze feature extractor $\mathcal{B}$ to capture appearance cues such as head pose and gaze orientation, while the head bounding box is projected into the embedding space to encode spatial position and scale. These cues are combined into a location-aware gaze vector $V_{gaze} \in \mathbb{R}^D$, which provides a compact representation for feature modulation. Compared with 2D gaze feature maps, this global vector reduces dimensionality, computational cost, and overfitting risk. Specifically, $V_{gaze}$ is fed into two parallel

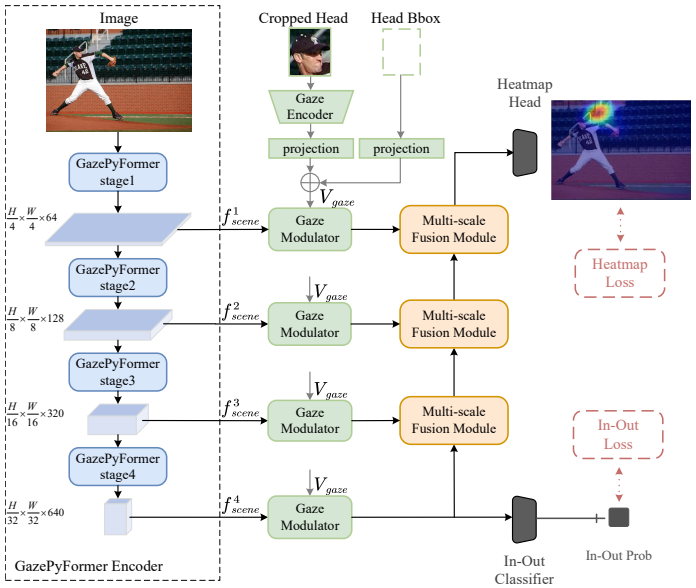


Fig. 1. Overview of the proposed GazePyFormer architecture. The GazePyFormer Encoder extracts multi-scale scene features from the input image, while the head branch produces gaze vectors. Scene and gaze features are fused by the GazeModulator and subsequently processed by a multi-scale fusion module. The resulting representations are used for in/out-of-frame classification and gaze heatmap regression.

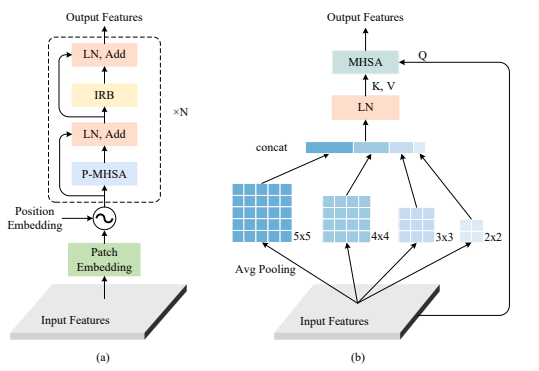


Fig. 2. Details of (a) the GazePyFormer Encoder block and (b) P-MHSA.

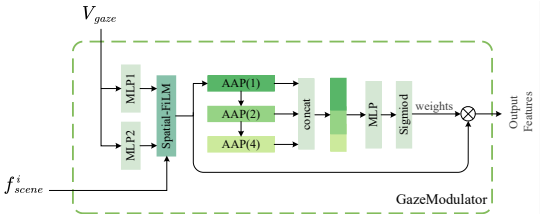


Fig. 3. Details of the GazeModulator.

multilayer perceptrons (MLPs) to generate modulation components $B_1$ and $B_2$, which are aligned with the feature maps for spatially consistent modulation, as formulated below:

$$B_1 = MLP1\left(V_{gaze}\right), B_2 = MLP2\left(V_{gaze}\right), \tag{7}$$

where $B_1, B_2 \in \mathbb{R}^{C\times H\times W}$. These components modulate the image features $f^i_{scene} \in \mathbb{R}^{C\times H\times W}$ via a spatial feature wise linear modulation layer (Spatial-FiLM) that enables fine-grained, position specific modulation, allowing the model to more effectively integrate gaze priors with spatial scene features and capture complex gaze-scene interactions. This operation is defined as:

$$X_{mod} = f^i_{scene} \odot (1 + B_1) + B_2, \tag{8}$$

yielding the modulated feature map $X_{mod} \in \mathbb{R}^{C\times H\times W}$. Next, $X_{mod}$ is processed by several adaptive average pooling (AAP) layers at different scale, enabling the model to capture diverse spatial dependencies and thus facilitating more accurate and robust gaze localization. Its expression is:

$$Y_i = AdaptiveAvgPool\left(X_{mod}, r_i\right), \tag{9}$$

where each $Y_i$ $(i = 1, 2, 3)$ denotes a pooled feature map with pooling ratio $r_i$. The pooled features $Y_i$ are concatenated along the channel dimension and fed into another MLP to generate spatial attention weights. This adaptive weighting enables the network to emphasize gaze-relevant regions by learning the importance of features at different scales. The operation is defined as:

$$W = \sigma\left(\mathrm{ReLU}\left(MLP\left(concat\left(Y_1, Y_2, Y_3\right)\right)\right)\right), \tag{10}$$

where $\sigma\left(\right)$ denotes the sigmoid activation function and $W \in \mathbb{R}^{C\times 1\times 1}$. Finally, these attention weights modulate the previously adjusted feature map $X_{mod}$ channel wise. The final output$Z \in \mathbb{R}^{C\times H\times W}$ is computed as:

$$Z = W \odot X_{mod}, \tag{11}$$

This adaptive modulation selectively enhances gaze-relevant features while suppressing irrelevant noise across spatial scales.

### D. Multi-scale Fusion Module

Inspired by the Conditional DPT Decoder in [22], we design a multi-scale fusion module to integrate encoder features across scales. At each decoder stage, the upsampled output from the previous layer is fused with the convolutionally processed feature map from the current encoder stage through element-wise addition, followed by convolutional refinement, upsampling, and dimensional projection. This coarse-to-fine fusion progressively restores spatial resolution, enhances contextual consistency, and facilitates accurate gaze heatmap generation.

### E. Loss Function

GazePyFormer is trained end-to-end using a weighted combination of two objectives:

**Heatmap loss** $L_{hm}$ is the mean squared error (MSE) between the predicted and ground-truth gaze heatmaps.

$$L_{hm} = \Sigma_{x,y}^{W_{hm},H_{hm}} \left\| A_{x,y}^{gt} - A_{x,y}^{pred} \right\|_2^2, \tag{12}$$

**In-vs-out loss** $L_{io}$ is the binary cross-entropy loss for predicting whether the gaze lies inside or outside the image frame.

$$\mathcal{L}_{\text{io}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log \sigma(p_i) + (1-y_i)\log(1-\sigma(p_i))\right], \tag{13}$$

The overall loss is expressed as:

$$L_{total} = \lambda_{hm} L_{hm} + \lambda_{io} L_{io}, \tag{14}$$

where $\lambda_{hm}$ and $\lambda_{io}$ are weighting coefficients that balance the two losses.

## III. Experiments

### A. Datasets

**GazeFollow** [5] is a large scale gaze following dataset with about 122k images and 130k annotated individuals, each labeled with head bounding boxes and 2D gaze points. Collected from multiple public datasets, it targets in-frame gaze and is evaluated via gaze heatmap regression.

**VideoAttentionTarget** [4] is a dataset based on videos consisting of 1,331 clips from 50 TV shows, with approximately 164k annotated instances over 71k frames. Each instance provides a head bounding box, a 2D gaze point, and an in/out-of-frame label, supporting both gaze heatmap regression and in/out-of-frame classification for comprehensive gaze following evaluation.

### B. Setup

**Data Preprocessing** Input images and head crop regions are standardized to 224×224. Head bounding boxes are slightly expanded, squared, and normalized to [0,1], after which corresponding head crops are extracted. Ground-truth gaze points are mapped to 64×64 heatmaps: a Gaussian kernel with $\sigma = 3$ is applied when $inout = 1$, otherwise a zero map is used.

**Training** The gaze backbone $\mathcal{B}$ adopts a ResNet-18 model pretrained on the large-scale Gaze360 dataset [26], which provides strong prior knowledge about human gaze behavior. The model then trained for 25 epochs using the AdamW optimizer, with an initial learning rate of 2.5e-5 and cosine annealing decay. The overall loss combines gaze heatmap regression ($\lambda_{hm} = 1000$) and in/out-of-frame classification ($\lambda_{io} = 10$), emphasizing precise gaze localization while maintaining spatial awareness.

**Validation** Since GazeFollow and VideoAttentionTarget lack official validation splits, we follow [22] by reserving 4,499 and 6,726 samples for validation, respectively.

**Evaluation Metrics** Model performance is evaluated using multiple complementary metrics. AUC measures the reliability of predicted gaze heatmaps, while the Euclidean L2 distance (Dist.) between the predicted peak and ground-truth gaze point is reported. For GazeFollow, both minimum and average distances across multiple annotators (Min. Dist. and Avg. Dist.) are used to reflect the variation among evaluators. For VideoAttentionTarget, Average Precision (AP) is adopted to evaluate in-frame versus out-of-frame gaze classification.

### C. Experimental Results

Table I compares GazePyFormer with existing state-of-the-art methods on GazeFollow and VideoAttentionTarget. For a fair evaluation, we categorize these methods into two groups: single-modality and multimodal approaches (utilizing depth, objects, or 3D pose).

**Performance in Image (I) Category**. Within the single-modality group, GazePyFormer achieves state-of-the-art performance across all primary metrics. On the GazeFollow dataset, it attains an AUC of 0.951 and reduces the minimum distance to 0.051, representing a 10.5% relative improvement over the previous best Transformer-based method, Sharingan [22]. This result indicates that the proposed hierarchical pyramid structure is more effective at resolving spatial ambiguities than fixed-resolution Transformers.

**Comparison with Multimodal Methods**. GazePyFormer remains highly competitive even against multimodal methods using auxiliary inputs. It outperforms SocialFusion [23] on AUC and achieves comparable results to GTPU3S [12]. These results suggest that the proposed GazeModulator can effectively capture geometric and semantic cues directly from RGB inputs, enabling a simpler and more efficient framework without sacrificing localization accuracy.

**Generalization on Video Data**. Similar trends are observed on VideoAttentionTarget. Despite processing frames independently without temporal modeling, our model maintains a lead over most single-frame counterparts and stays competitive with complex video-based frameworks. The consistency of these results across different datasets and annotation protocols validates the robustness of our multi-scale fusion strategy in capturing diverse gaze-to-target spatial relationships.

### D. Ablation Study

We conducted a series of ablation experiments on GazeFollow to examine the contribution of each design component.

**Encoder Architecture** We compare different encoder backbones by replacing our encoder with alternative architectures. Table II shows that Transformers consistently outperform CNNs,

TABLE I

RESULTS OF OUR GAZEPYFORMER ARCHITECTURE ON THE GAZEFOLLOW AND VIDEOATTENTIONTARGET DATASETS. THE BEST SCORES FOR MODELS ARE GIVEN IN BOLD, WHILE THE BEST SCORES IN GENERAL ARE UNDERLINED. THE MODALITY COLUMN USES THE CODES I (IMAGE), T (TIME), D (DEPTH), P (POSE), T(TEXT) AND O (OBJECT).

| **Method** | **Modality** | **GazeFollow** | | | **VideoAttentionTarget** | | |
|---|---|---|---|---|---|---|---|
| | | **AUC↑** | **Avg.D.↓** | **Min.D.↓** | **AUC↑** | **Dist.↓** | **AP↑** |
| VSG-IA [27] | I+D+O | 0.923 | 0.128 | 0.069 | 0.880 | 0.118 | 0.881 |
| PDP [28] | I+D | 0.934 | 0.123 | 0.065 | 0.917 | 0.109 | **0.908** |
| MM-Gaze [29] | I+D+P | 0.943 | 0.114 | <u>0.056</u> | 0.913 | 0.110 | 0.879 |
| MTGS [21] | I+T | 0.929 | 0.116 | 0.059 | – | <u>0.105</u> | 0.869 |
| SPGCG [13] | I+D | 0.936 | 0.121 | 0.061 | 0.918 | 0.108 | 0.882 |
| TSGTD [30] | I+O | – | <u>0.108</u> | **0.051** | – | – | – |
| SocialFusion [23] | I+D+O | 0.940 | 0.128 | 0.065 | – | – | – |
| GTPU3S [12] | I+D | <u>0.949</u> | 0.112 | 0.058 | **0.931** | **0.101** | 0.869 |
| Depth-Aware Gaze [6] | I | 0.919 | 0.126 | 0.076 | 0.881 | 0.134 | 0.880 |
| HGTTR [18] | I | 0.917 | 0.133 | 0.069 | 0.904 | 0.126 | 0.854 |
| GTAVP [20] | I | 0.928 | 0.133 | 0.071 | 0.862 | 0.160 | 0.841 |
| Sharingan [22] | I | 0.944 | 0.113 | 0.057 | – | 0.107 | 0.891 |
| **GazePyFormer** | I | **0.951** | **0.107** | **0.051** | <u>0.930</u> | <u>0.105</u> | <u>0.898</u> |

TABLE II

ANALYSIS OF ENCODER SETTINGS ON THE GAZEFOLLOW DATASET.

| **Encoder Setting** | **AUC↑** | **Avg.D.↓** | **Min.D.↓** |
|---|---|---|---|
| ResNet50 | 0.921 | 0.136 | 0.078 |
| ViT | 0.943 | 0.112 | 0.056 |
| Swin Transformer | 0.940 | 0.117 | 0.059 |
| GazePyFormer (Ours) | **0.951** | **0.107** | **0.051** |

TABLE III

ANALYSIS OF ENCODER STAGES ON THE GAZEFOLLOW DATASET. A CHECK MARK INDICATES THAT THE CORRESPONDING ENCODER STAGE IS USED.

| **Stage of Encoder** | | | | **AUC↑** | **Avg.D.↓** | **Min.D.↓** |
|---|---|---|---|---|---|---|
| **1** | **2** | **3** | **4** | | | |
| ✓ | | | | 0.635 | 0.311 | 0.228 |
| ✓ | ✓ | | | 0.827 | 0.196 | 0.125 |
| ✓ | ✓ | ✓ | | 0.944 | 0.109 | 0.053 |
| ✓ | ✓ | ✓ | ✓ | **0.951** | **0.107** | **0.051** |

TABLE IV

ANALYSIS OF POOLING STYLE SELECTION ON THE GAZEFOLLOW DATASET.

| **Pooling Setting** | **AUC↑** | **Avg.D.↓** | **Min.D.↓** |
|---|---|---|---|
| Average Pooling | **0.951** | **0.107** | **0.051** |
| Max Pooling | 0.949 | 0.109 | 0.052 |
| Learnable Hybrid Pooling (Avg + Max) | 0.947 | 0.110 | 0.054 |
| Convolutional Pooling | 0.946 | 0.112 | 0.055 |

TABLE V

ANALYSIS OF MULTIPLE PYRAMID POOLING LAYERS IN THE FIRST BLOCK ON THE GAZEFOLLOW DATASET. A CHECK MARK INDICATES THAT THE CORRESPONDING POOLING RATIO IS USED.

| **Pooling Ratio** | | | | **AUC↑** | **Avg.D.↓** | **Min.D.↓** |
|---|---|---|---|---|---|---|
| **12** | **16** | **20** | **24** | | | |
| ✓ | | | | 0.945 | 0.112 | 0.056 |
| ✓ | ✓ | | | 0.947 | 0.111 | 0.055 |
| ✓ | ✓ | ✓ | | 0.948 | 0.110 | 0.055 |
| ✓ | ✓ | ✓ | ✓ | **0.951** | **0.107** | **0.051** |

demonstrating the advantage of self-attention in modeling long-range head–scene dependencies. CNN baselines use features from different ResNet50 stages, while ViT-based models use representations from four Transformer blocks. Based on this, GazePyFormer employs a hierarchical, gaze-aware encoder for more accurate heatmap prediction.

**Encoder Stages** To assess the contribution of each hierarchical stage in our encoder, we perform an ablation study by progressively varying the depth of the encoder. As shown in Table III, increasing the number of encoder stages progressively improves gaze prediction performance, indicating that lower layers capture fine spatial details while higher ones encode semantic context. Gains plateau after the third stage, suggesting a balance between detail and abstraction.

**Pooling Type** We compare four pooling strategies in the GazePyFormer encoder: average pooling, max pooling, learnable hybrid pooling, and convolution. Table IV shows only minor performance differences, with average pooling giving slightly more stable and generalizable results. Since max and hybrid pooling may suppress subtle contextual cues, and convolution adds parameters without clear gains, we adopt average pooling in GazePyFormer.

**Multiple Pyramid Pooling Layers** We study the effect of multiple pyramid pooling layers by varying only their number in the first backbone stage. Table V shows that increasing the number of pooling layers consistently improves performance, highlighting the importance of early-stage multi-scale context. The best results are obtained when all pooling ratios are used, indicating that richer multi-scale representations improve gaze localization and robustness to head size and scene layout variations.

**Optimal Pooling Combination** We compare different adaptive pooling size combinations in the GazeModulator. Table VI shows that balanced multi-scale settings, such as [1,2,4], perform best by capturing both local and mid-scale context, whereas larger-scale combinations may miss subtle spatial cues.

### *E. Visualization*

We evaluate GazePyFormer using visualization on the GazeFollow test set. Predicted heatmaps, gaze points, and ground truth are overlaid with head bounding boxes for qualitative comparison. As shown in Fig. 4, predictions closely match the

TABLE VI
ANALYSIS OF POOLING SIZE COMBINATION SELECTION IN THE GAZEMODULATOR ON THE GAZEFOLLOW DATASET.

| Pooling Size Combination | AUC↑ | Avg.D.↓ | Min.D.↓ |
|---|---|---|---|
| [1, 2, 3] | 0.946 | 0.109 | 0.054 |
| [1, 2, 4] | **0.951** | **0.107** | **0.051** |
| [2, 4, 8] | 0.943 | 0.112 | 0.057 |

Fig. 4. Visualization of predicted gaze maps and ground truth. The background heatmap illustrates the predicted gaze distribution, where the magenta point denotes the predicted gaze target and the yellow point indicates the ground-truth gaze position.

ground truth across diverse scenarios, indicating accurate gaze localization.

Next, Fig. 5 shows attention maps from four hierarchical Transformer layers together with the final predicted gaze heatmap. Attention shifts from global scene perception to gaze-relevant regions, and finally back to broader semantic context for heatmap regularization. This global to local to global process supports accurate gaze localization and in/out-of-frame reasoning, further demonstrating the effectiveness of the GazePyFormer Encoder and GazeModulator.

Finally, we assess the generalization ability of GazePyFormer on previously unseen images outside the training and test sets. Head regions are detected by a YOLOv5m model [31] pretrained on CrowdHuman [32], and all inputs are resized to 224×224 before inference. As shown in Fig. 6, the model produces visually plausible gaze predictions across diverse scenarios, indicating strong transferability beyond the training data.

### *F. Limitation and Discussion*

Fig. 7 presents representative failure cases, mainly involving head occlusion, out-of-frame targets, and multi-person scenes with ambiguous gaze cues. Performance further declines in crowded environments due to overlapping context and complex social interactions. These results indicate that GazePyFormer is still limited in visually ambiguous scenarios. Future work will focus on multi-person relational reasoning and context-aware modeling.

Fig. 5. Visualization of attention maps after gaze vector modulation across four stages.

Fig. 6. Demonstration of gaze-follow testing on unseen images.

## IV. CONCLUSION

In this paper, we introduced GazePyFormer, a hierarchical Transformer framework designed to address the scale-variance challenges in gaze following. By embedding pyramid pooling directly into the encoding stages, our model maintains a continuous spatial hierarchy that effectively bridges the gap between local head cues and distant scene targets. The integration of the GazeModulator further ensures that gaze-specific priors are adaptively aligned with these multi-scale features via spatial modulation. Empirical results on GazeFollow and VideoAttentionTarget demonstrate that GazePyFormer sets new benchmarks for RGB-only gaze following, achieving high precision even in cluttered environments without requiring auxiliary modalities.

While our approach shows strong performance on static frames, it does not yet fully exploit the temporal dynamics inherent in video sequences. Future research will focus on extending this hierarchical architecture to include temporal self-attention and exploring its potential in modeling complex multi-person social interactions in real-time.


## ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grants 62366006 and 62366005, by the Guangxi Key Laboratory of Precision Navigation Technology and Application under Project DHKL2417, by the 2025 Joint Cultivation Project of the National Natural Science Foundation under Guangxi Normal University, and by the Guangxi Natural Science Foundation for Youths under Grant 2026GXNSFBA00640143.

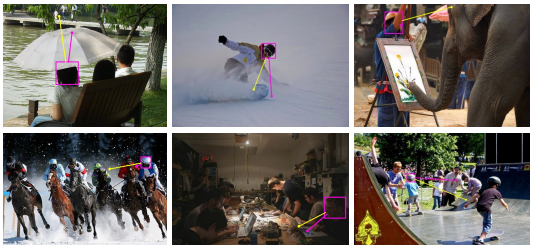

Fig. 7. Failure cases observed during the visualization process.

## REFERENCES


[1] P. Prasse, D. R. Reich, S. Makowski, T. Scheffer, and L. A. Jäger, "Improving cognitive-state analysis from eye gaze with synthetic eye-movement data," *Computers & Graphics*, vol. 119, p. 103901, 2024.

[2] G. R. Chhimpa, A. Kumar, S. Garhwal, F. Khan, Y.-K. Moon *et al.*, "Revolutionizing gaze-based human-computer interaction using iris tracking: A webcam-based low-cost approach with calibration, regression and real-time re-calibration," *IEEE Access*, 2024.

[3] J. Gong, S. Cao, S. Korivand, and N. Jalili, "Reconstructing human gaze behavior from eeg using inverse reinforcement learning," *Smart Health*, vol. 32, p. 100480, 2024.

[4] E. Chong, Y. Wang, N. Ruiz, and J. M. Rehg, "Detecting attended visual targets in video," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5396–5406.

[5] A. Recasens, A. Khosla, C. Vondrick, and A. Torralba, "Where are they looking?" *Advances in neural information processing systems*, vol. 28, 2015.

[6] T. Jin, Q. Yu, S. Zhu, Z. Lin, J. Ren, Y. Zhou, and W. Song, "Depth-aware gaze-following via auxiliary networks for robotics," *Engineering Applications of Artificial Intelligence*, vol. 113, p. 104924, 2022.

[7] H. Xia, Z. Gong, Y. Tan, and S. Song, "Joint pyramidal perceptual attention and hierarchical consistency constraint for gaze estimation," *Computer Vision and Image Understanding*, vol. 248, p. 104105, 2024.

[8] B. Wang, T. Hu, B. Li, X. Chen, and Z. Zhang, "Gatector: A unified framework for gaze object prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 588–19 597.

[9] A. Saran, S. Majumdar, E. S. Short, A. Thomaz, and S. Niekum, "Human gaze following for human-robot interaction," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 8615–8621.

[10] D. Lian, Z. Yu, and S. Gao, "Believe it or not, we know what you are looking at!" in *Asian Conference on Computer Vision*. Springer, 2018, pp. 35–50.

[11] G. Zeng, J. Wang, Z. Xu, P. Yin, W. Ren, D. Xie, and J. Zhu, "Gaze label alignment: Alleviating domain shift for gaze estimation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 9, 2025, pp. 9780–9788.

[12] L. Gao, F. Sun, and Y. Liu, "Gaze target prediction with the understanding of 3d scenes," in *Chinese Conference on Image and Graphics Technologies*. Springer, 2025, pp. 129–143.

[13] F. Liu, K. Li, Z. Zhong, W. Jia, B. Hu, X. Yang, M. Wang, and D. Guo, "Depth matters: Spatial proximity-based gaze cone generation for gaze following in wild," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 20, no. 11, pp. 1–24, 2024.

[14] Z. Liang, J. Liu, Y. Guan, and J. Rojas, "Visual-semantic graph attention networks for human-object interaction detection," in *2021 IEEE international conference on robotics and biomimetics (ROBIO)*. IEEE, 2021, pp. 1441–1447.

[15] S. Qi, W. Wang, B. Jia, J. Shen, and S.-C. Zhu, "Learning human-object interactions by graph parsing neural networks," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 401–417.

[16] A. Gupta, P. Vuillecard, A. Farkhondeh, and J.-M. Odobez, "Exploring the zero-shot capabilities of vision-language models for improving gaze following," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 615–624.

[17] Y. Song, X. Wang, J. Yao, W. Liu, J. Zhang, and X. Xu, "Vitgaze: gaze following with interaction features in vision transformers," *Visual Intelligence*, vol. 2, no. 1, pp. 1–15, 2024.

[18] D. Tu, X. Min, H. Duan, G. Guo, G. Zhai, and W. Shen, "End-to-end human-gaze-target detection with transformers," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 2192–2200.

[19] C. Nakatani, H. Kawashima, and N. Ukita, "Interaction-aware joint attention estimation using people attributes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 10 224–10 233.

[20] T. Huang and J. Huang, "Gaze target detection with visual prompt tuning based on attention," in *International Conference on Artificial Neural Networks*. Springer, 2024, pp. 415–429.

[21] A. Gupta, S. Tafasca, A. Farkhondeh, P. Vuillecard, and J.-M. Odobez, "A novel framework for multi-person temporal gaze following and social gaze prediction," *arXiv preprint arXiv:2403.10511*, 2024.

[22] S. Tafasca, A. Gupta, and J.-M. Odobez, "Sharingan: A transformer architecture for multi-person gaze following," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 2008–2017.

[23] H. Tahboub, W. Shi, G. Hua, and H. Jiang, "Socialfusion: Addressing social degradation in pre-trained vision-language models," *arXiv preprint arXiv:2512.01148*, 2025.

[24] Y.-H. Wu, Y. Liu, X. Zhan, and M.-M. Cheng, "P2t: Pyramid pooling transformer for scene understanding," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 11, pp. 12 760–12 771, 2022.

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[26] P. Kellnhofer, A. Recasens, S. Stent, W. Matusik, and A. Torralba, "Gaze360: Physically unconstrained gaze estimation in the wild," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6912–6921.

[27] Z. Hu, K. Zhao, B. Zhou, H. Guo, S. Wu, Y. Yang, and J. Liu, "Gaze target estimation inspired by interactive attention," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 12, pp. 8524–8536, 2022.

[28] Q. Miao, M. Hoai, and D. Samaras, "Patch-level gaze distribution prediction for gaze following," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 880–889.

[29] A. Gupta, S. Tafasca, and J.-M. Odobez, "A modular multimodal architecture for gaze target prediction: Application to privacy-sensitive settings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5041–5050.

[30] S. Tafasca, A. Gupta, V. Bros, and J.-M. Odobez, "Toward semantic gaze target detection," *Advances in neural information processing systems*, vol. 37, pp. 121 422–121 448, 2024.

[31] G. Jocher, A. Chaurasia, J. Qiu *et al.*, "Yolov5 by ultralytics," https://github.com/ultralytics/yolov5, 2020.

[32] S. Shao, Z. Li, T. Zhang, C. Peng, G. Yu, X. Zhang, and J. Sun, "Crowdhuman: A benchmark for detecting human in a crowd," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.