# Object-aware Gaze Target Detection

Francesco Tonini[1,2], Nicola Dall'Asen[1,3], Cigdem Beyan[1], Elisa Ricci[1,2]
[1] University of Trento, Trento, Italy [2] Fondazione Bruno Kessler, Trento, Italy
[3] University of Pisa, Pisa, Italy
{francesco.tonini, nicola.dallasen, cigdem.beyan, e.ricci}@unitn.it

## Abstract

*Gaze target detection aims to predict the image location where the person is looking and the probability that a gaze is out of the scene. Several works have tackled this task by regressing a gaze heatmap centered on the gaze location, however, they overlooked decoding the relationship between the people and the gazed objects. This paper proposes a Transformer-based architecture that automatically detects objects (including heads) in the scene to build associations between every head and the gazed-head/object, resulting in a comprehensive, explainable gaze analysis composed of: gaze target area, gaze pixel point, the class and the image location of the gazed-object. Upon evaluation of the in-the-wild benchmarks, our method achieves state-of-the-art results on all metrics (up to 2.91% gain in AUC, 50% reduction in gaze distance, and 9% gain in out-of-frame average precision) for gaze target detection and 11-13% improvement in average precision for the classification and the localization of the gazed-objects. The code of the proposed method is available here.*

## 1. Introduction

Gazing is a powerful nonverbal signal, which indicates the visual attention of a person and allows one to understand the interest, intention, or (future) action of people [12]. For this reason, gaze analysis has widely been used in several disciplines such as human-computer interaction [27, 37], neuroscience [8, 29], social and organizational psychology [3, 11], and social robotics [1] to name a few.

Even though human beings have a remarkable capability to decode the gaze behavior of others, realizing this task *automatically* remains a challenging problem [2, 34, 35]. The computer vision community has tackled the automated gaze behavior analysis in terms of two tasks: (a) *gaze estimation* and (b) *gaze target detection*. Gaze estimation stands for predicting the person's gaze direction (usually in 3D) when typically a *cropped human head image* is given as the input [4, 14, 15, 21]. Instead, gaze target detection (also referred to as gaze-following) is to determine the specific (2D or 3D) location that a human is looking at in an *in-the-wild* scene [7, 13, 18].

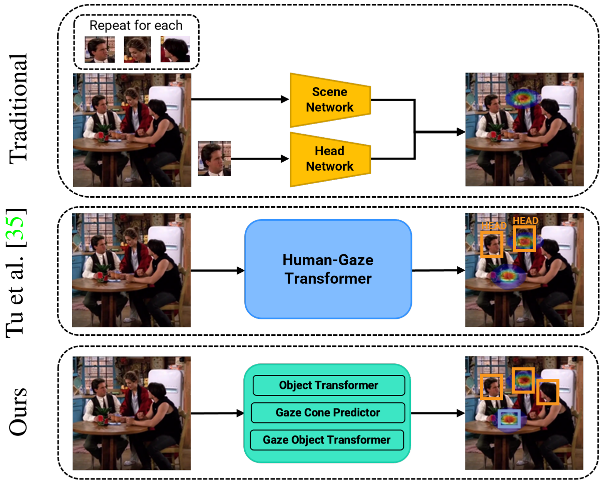


Figure 1: The overall methodology of the existing approaches and ours.

Several works utilize head pose features and the saliency maps of possible gaze targets to perform gaze target detection. For instance, [6, 23, 30, 31] followed a two-pathway learning scheme, where one path learns feature embeddings from the scene image, and the other path models the head crops belonging to the person whose gaze target is aimed to be predicted. Chong *et al.* [7] extend the aforementioned two-pathway approach to perform spatio-temporal modeling to determine the gaze targets in videos. In the same vein, a few other methods exist: [2, 13, 19, 26, 34]. Among them, some consider a third path to model the depth map of the scene image, which is determined by a monocular depth estimator [13, 19, 26, 34]. Differently, others [2] inject depth maps and 2D-human poses to improve the 3D understanding of the scenes, resulting in better gaze target detection. The results achieved by these approaches (referred to as *traditional methods* throughout the manuscript, see Fig. 1-top) are highly remarkable since they demonstrated that gaze target estimation could be directly performed on images or videos in contrast to using low-intrusive wearable eye trackers, which notoriously have several issues in terms of cost, battery life, and calibration. On the other hand, traditional methods also have some major drawbacks. First, both training and inference require carefully human-annotated head crops. Therefore, to ensure that traditional methods work well in real-life practical applications, there is a need for additional and highly accurate head detectors. Indeed, Tu *et al.* [35] showed notable performance drops of traditional methods when head detectors were involved instead of using manually annotated head locations. A second limitation concerns the fact that traditional methods can perform a single gaze target detection at a time; thus, for scenes containing multiple people, the models should be run repeatedly for each person. Besides the computational complexity such implementation brings in, post-processing is also needed to combine the detected gaze targets of different subjects in the same scene. Tu *et al.* [35] to some extent overcome the shortcomings mentioned above by introducing a Transformer-based architecture that explicitly learns how to detect and localize the head during gaze target detection (see Fig. 1-middle). However, the contribution of objects to decipher the human-human/object gazing is completely omitted in [35].

Several studies show that people typically gaze at living or non-living *objects* in the scene during social and physical interactions [5, 22, 25, 33, 36, 38, 39]. Motivated by this, we pursue an object-aware gaze target detection, instead of using features extracted from holistic scene images and head crops as in traditional methods: [2, 6, 7, 13, 19, 23, 26, 30, 34] or learning how to detect and localize the head of the person-in-interest (the one whose gaze target to be detected) as in [35]. Our proposal is not only able to predict the *gaze area* (in terms of heatmaps) that people looking at and determine if the gaze target is inside or outside of the scene but also *localize the objects* and *predicts the objects' classes* (including head) on which the gaze point is (see Fig. 1-bottom). The further has significant practical usage since it brings in an *explainable* gaze analysis (see Table 1 for details).

The proposed method is an end-to-end Transformer-based architecture. Given a scene image, we first extract all objects, including the ones classified as heads, with an *Object Detector Transformer*. Then, for each head, a gaze vector is predicted. Using this gaze vector, we build a *gaze cone* for each person individually, allowing the model to filter out the objects that are not in a person's Field of View (FoV). Subsequently, a masked transformer (called *Gaze Object Transformer*) learns the interactions between the detected heads and objects, boosting the gaze target detection performance in terms of both heatmaps and gaze points (*i.e.* a single pixel in the scene). Furthermore, this architecture has a remarkable capability to predict whether a gaze target point is out of the frame. The extensive evaluations on two large-scale benchmark datasets show the superior performance of our method *w.r.t.* state-of-the-art (SOTA) gaze target detectors. At the same time, our model has additional competence to accurately predict the gazed objects' locations and the associated classes as empirically demonstrated. The ablation study highlights the importance of all components and specifically the needs for our main technical contributions, *i.e.* the Gaze Cone Predictor and the Gaze Object Transformer.

| Method | Wout/ Head Loc. Given | Multiple People | Head Detection | Object Localization | Object Classification |
|---|---|---|---|---|---|
| Traditional | ✗ | ✗ | ✗ | ✗ | ✗ |
| Tu *et al.* [35] | ✓ | ✓ | ✓ | ✗ | ✗ |
| Ours | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 1: Existing gaze target detection methods compared to ours. Ours is more explainable since, for every person in the scene, it can detect the object class and bounding box location on which the gaze is. It learns the scene objects (including the head) without using the head locations supplied by datasets.

To summarize: (1) We introduce a novel object-oriented gaze target detection method. (2) This end-to-end Transformer-based model automatically detects the heads and other objects in the scene to build associations between every head and the gazed-head/object, resulting in a comprehensive, explainable gaze analysis composed of: gaze target area, gaze pixel point, the class of gazed-object, the bounding box of the gazed-object as well as predicting whether the gazed point is out of the frame. (3) We demonstrate SOTA results on standard datasets regarding all evaluation metrics for gaze-target detection (up to 2.91% gain in AUC, 50% reduction in gaze distance, and 9% gain in out-of-frame AP), gazed-object classification and localization (11-13% gain in AP) and in case of low/high variance across gaze annotations. (4) The code of the proposed method is publicly available (see Supp. Mat. for review, the link will be inserted in camera-ready). We also release our implementation for [35] since during our private communications with the authors, we are informed that their code at the moment cannot be shared by them due to their ongoing collaborations with a company.

## 2. Related Work

The main focus of this paper is to determine the location that a human is looking at in an in-the-wild scene captured from the third-person view. To this end, Recasens *et al.* [30] presented the first relevant dataset called GazeFollow, and proposed a two-branch Convolutional Neural Network (CNN) whose first branch estimates the saliency from scene images and the second branch processes manually annotated head crops together with their location information. Several subsequent works [2, 7, 13, 19, 23, 31, 34] adopted this two-branch structure and further introduced additional components. For instance, Chong *et al.* [6] brought in de-

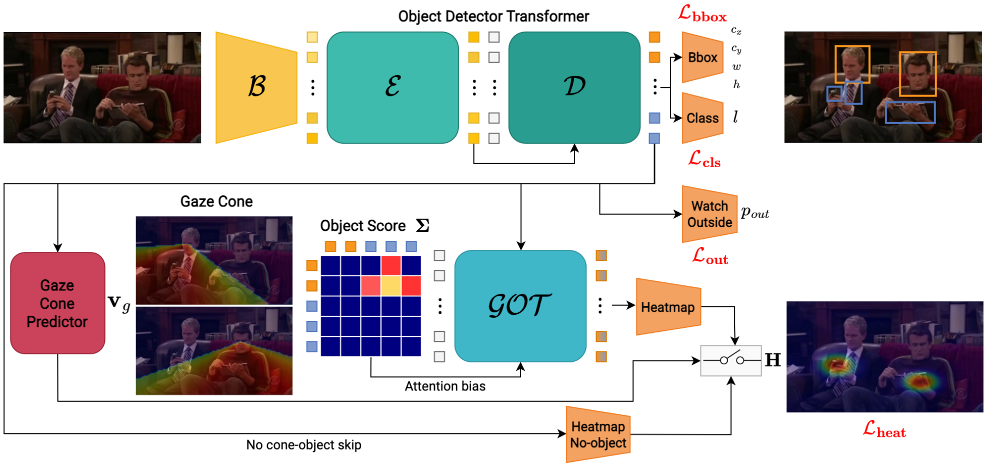


Figure 2: **Proposed method.** The encoder ($\mathcal{E}$) and decoder ($\mathcal{D}$) of the Object Detector Transformer operate on the features extracted by a backbone $\mathcal{B}$ to learn rich object features used to detect and localize objects (including heads) in the scene. Head features are used to build the **gaze cone**. Objects in the cone are extremely likely to be gaze-interesting. The *object score* matrix $\Sigma$ boosts attention scores in the *Gaze Object Transformer* ($\mathcal{GOT}$), whose output features are used to build the gaze heatmap. If no object lies in the cone, a skip-connection lets the network predict the heatmap from head features only.

tecting the gaze targets not-being in the scene (so-called *out-of-frame* detection). Later on, the same authors [7] integrated a CNN-LSTM into their pipeline, modeling the dynamics of gaze in *videos* and making frame-based inferences. They also introduced the VideoAttentionTarget dataset, made of videos. A few studies incorporated the depth maps obtained from monocular depth estimators in addition to the embeddings learned from RGB scenes and head crops [13, 19, 26, 34]. Fang *et al.* [13] integrated the precise detection of head pose and the location of eyes. Jin *et al.* [19] used two auxiliary networks, one to learn depth features and the other to compute 3D-gaze orientation features. However, detecting head pose, eye locations, etc., are already challenging tasks to perform in-the-wild, and their inaccurate results can affect gaze target detection negatively. A better solution could be leveraging the collaborative learning of scene, depth, and head features, as shown in [34]. Bao *et al.* [2] proposed a method taking an RGB image and a head crop at a time and further using the depth map and a 2D human body pose detector to reconstruct the 3D scene with point clouds. Such a model [2] requires several detectors to be fine-tuned and therefore increases the computational complexity. Furthermore, it underperforms compared to, e.g., [13, 34] on the VideoAttentionTarget dataset. Qiaomu *et al.* [26] used the same modalities as [34] but also included a temporal attention model and replaced the in/out prediction encoder of [7] with a patch distribution prediction module, resulting in effective performance in case of large annotation variances. Unlike aforesaid approaches, aka *traditional methods*, using pretrained CNN backbones, Tu *et al.* [35] introduced the first Transformer-based approach, outperforming the others. Drastic performance drops for traditional methods were also demonstrated in [35], when they were evaluated with the head locations predicted by automated head detectors.

We stand out from the prior art as our method performs simultaneous gaze target detection of multiple persons in the scene by mutually learning localization and classification of the gazed-objects (including the head) and determining the head-head/object gaze interactions. Our end-to-end Transformer-based model explicitly aims to provide explainable gaze target detection, which has not been accomplished before (see Table 1 for comparisons).

## 3. Method

The proposed method is shown in Fig 2. Given an image, we first predict the set of objects $\mathbf{O} = \{(c_x, c_y, w, h, l)\}$ in it, where $(c_x, c_y, w, h)$ represent the center coordinates of a single object and its width and height, respectively, $l \in [0, CLS)$ is an object's label, and $CLS$ is the number of classes, including a special *no object* ($\emptyset$) class (described in Sec. 4.2). To this end, after extracting the image features through a backbone $\mathcal{B}$, we use an *Object Detector Transformer* that reasons on the scene features with the encoder $\mathcal{E}$ and learns relevant object features with the decoder $\mathcal{D}$. Such features are used to differentiate between heads $\mathbf{O_h}$ and other objects in the scene. For each head $\mathbf{O}_h^i$, we feed its features to the *Gaze Cone Predictor* to determine a gaze vector $\mathbf{v}_g^i$ that represents the gaze direction of the person. This gaze vector is used to build a gaze cone with an angle of $\alpha$ corresponding to the Field of View (FoV) and to selectively maintain the objects that are inside the cone for each head. The *Gaze-Object Transformer* ($\mathcal{GOT}$) models the relationships between the detected objects and predicts the probability of them being the gaze target of any person, with a higher probability for the objects closer to the gaze vector. The gaze of each person is represented as a Gaussian heatmap $\mathbf{H}^i$ centered on the gaze point $\mathbf{p}_g^i$, and when no object is present inside the gaze cone, we use a *no cone-object skip* to compute the heatmap directly from the head features. We also use the head features to predict the probability of the gaze target being outside the frame. To sum up, our model consists of three major components: (a) Object Detector Transformer, (b) Gaze Cone Predictor, and (c) Gaze Object Transformer, which are described thoroughly in the following sections.

### 3.1. Object Detector Transformer

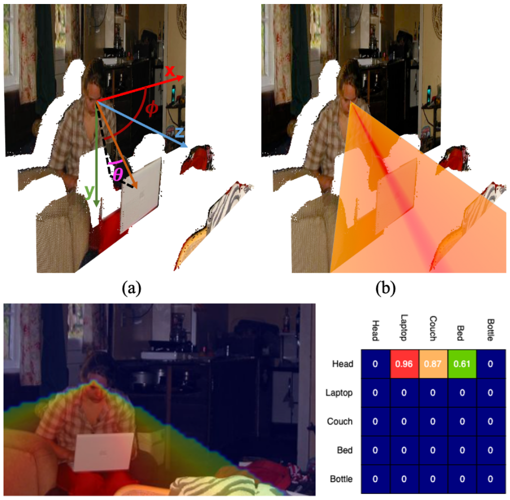
Given an RGB image $\mathbf{x} \in \mathbb{R}^{C \times H \times W}$, we aim to predict the bounding boxes and labels of objects. We start by extracting a feature map $\mathbf{f_b} \in \mathbb{R}^{C_b \times H_b \times W_b}$ with a convolutional backbone $\mathcal{B}$, and we linearly project the channel dimension to a lower space $C^{b'}$ due to the high channel dimensionality. We flatten the spatial dimensions and obtain $\mathbf{f_b}^{'} \in \mathbb{R}^{H_b W_b \times C_b^{'}}$, which is fed to a transformer encoder $\mathcal{E}$ that enhances the coarse image features extracted by $\mathcal{B}$.

$\mathcal{E}$ is designed as a stack of multi-head self-attention (MHSA) and feed-forward (FFN) layers. The projected output of $\mathcal{B}$, $\mathbf{f_b}^{'}$, forms the input queries $Q$, keys $K$, and values $V$ of $\mathcal{E}$. To retain the spatial information of the feature map, we add positional encodings for $Q$ and $K$. The output of the encoder, $\mathbf{f_e}$, forms the input $K$ and $V$ of the cross-attention module of the transformer decoder $\mathcal{D}$.

$\mathcal{D}$ completes our Object Detector Transformer and introduces a multi-head cross-attention module to obtain object-relevant features. First, the decoder performs self-attention on a set of learnable embeddings $\mathbf{e_d} \in \mathbb{R}^{N \times C_b^{'}}$, where $N$ is the maximum number of objects to be predicted. Similar to $\mathcal{E}$, we add the learnable embeddings $\mathbf{e_d}$ with a set of fixed positional embeddings. The output of the self-attention on $\mathbf{e_d}$ is then fed to a multi-head cross-attention module, where $\mathbf{e_d}$ are the queries, and $\mathbf{f_e}$ are the keys and values. The output features $\mathbf{f}_d$ of the transformer decoder are finally used by two multi-layer perceptrons (MLP) to predict the object bounding box (Bbox) and class, respectively.

### 3.2. Gaze Cone Predictor

The objects predicted by the *Object Detector Transformer* may appear in an area outside the FoV of a person but inside of another person(s). Since our method performs *multi-person* gaze prediction, we must consider the mentioned scenario and selectively focus on objects in the FoV of each individual. To this end, the *Gaze Cone Predictor* produces a gaze cone for each head detected and allows the $\mathcal{GOT}$ to focus on only the relevant objects on a person-by-person basis. For the objects detected as a head, the gaze cone, which can be either in 2D or 3D, is built based on the estimated gaze vector. The gaze cone allows us to build an *object score* matrix ($\Sigma$) based on the relationship between head-head/objects.



(a)

(b)

(c)

(d)

|  | Head | Laptop | Couch | Bed | Bottle |
|---|---|---|---|---|---|
| Head | 0 | 0.96 | 0.87 | 0.61 | 0 |
| Laptop | 0 | 0 | 0 | 0 | 0 |
| Couch | 0 | 0 | 0 | 0 | 0 |
| Bed | 0 | 0 | 0 | 0 | 0 |
| Bottle | 0 | 0 | 0 | 0 | 0 |

Figure 3: (a) Our 3D cone construction from the head center point; $\theta$ and $\phi$ are the polar and azimuthal angle, respectively. Exploded view computed with the depth map. (b) 3D gaze cone considers the depth and, in this way, excludes the objects unrelated to the gaze vector, in this case, the couch and the bed on the right. Instead, in (c) 2D gaze cone, the couch is inside it, although common sense would tell that it should not be. (d) *Object score* matrix $\Sigma$ of (c) when the detected objects are: *head*, *laptop*, *couch*, *bed*, and *bottle* classes.

In detail, an MLP takes as input the features of objects detected as heads $\mathbf{O}_h$ and estimates, for each of them, a 3D gaze vector $\mathbf{v}_g^i = (\theta^i, \phi^i, \rho^i)$. Each gaze vector uniquely identifies the orientation of the person's gaze with $\theta$, $\phi$, and $\rho$, which are the polar angle, azimuthal angle, and magnitude of the vector, respectively. For each gaze vector $\mathbf{v}_g^i$, we design a 3D cone of angle $\alpha$ and apex $(c_x^i,\ c_y^i,\ c_z^i)$ representing the FoV of a person, where $c_x^i$, $c_y^i$, and $c_z^i$ are the center coordinates of the head. The cone axis has the same direction as the gaze vector, and the intensity of the cone, *i.e.*, the point saliency, is calculated as the cosine similarity between $\mathbf{v}_g^i$ and all vectors inside the cone starting from $(c_x^i,\ c_y^i,\ c_z^i)$. In the 2D case, $\theta$ is not available, and we only have one angle $\phi$ and the magnitude $\rho$ for the gaze vector, while the 2D cone is still in the center of the apex but spans only in 2D instead of 3D. We adopt the discretized space of the same dimensionality of the predicted heatmap presented in [16], while we extend it to the 3D case, with $x$, $y$, and $z$ axis corresponding to the width, height, and depth of the image. For the 2D cone building, we follow the approach of [16], but we constrain the cone to be a fixed angle $\alpha$, which is in line with the FoV of human boundaries [17].

Below, the derivations are given for the 3D gaze cone, but the corresponding 2D implementation of them is the same except for not having the depth coordinates as described above. Refer to the visual explanation of the 2D and 3D gaze cone in Fig. 3a-c. Formally, let $angle(\mathbf{v}_a, \mathbf{v}_b)$ be the absolute value of the angle between two vectors, and $\sigma(\mathbf{v}_a, \mathbf{v}_b)$ be the cosine similarity between two vectors $\mathbf{v}_a$ and $\mathbf{v}_b$ conditioned on the cone angle $\alpha$:

$$\sigma(\mathbf{v}_a, \mathbf{v}_b) = \begin{cases} \cos(\mathbf{v}_a, \mathbf{v}_b) & \text{if } angle(\mathbf{v}_a, \mathbf{v}_b) \leq \frac{\alpha}{2}, \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The projected 3D gaze cone of a person $i$, $\mathbf{CD}^i_{3D}$, whose head center coordinates are $c^i_x, c^i_y, c^i_z$, and predicted gaze vector $\mathbf{v}^i_g$, is defined as:

$$\begin{gathered} \mathbf{CD}^i_{3D} = \{\sigma(\mathbf{v}^i_g, \mathbf{v}^{ijkl}_H)\} \\ \forall j, k, l \in [0, w) \times [0, h) \times [0, d) \end{gathered} \quad (2)$$

where $w$, $h$, and $d$ are the width, height, and depth of the space on which the 3D cone is computed, and $\mathbf{v}^i_H$ indicates the vectors in the discretized space starting from $(c^i_x, c^i_y, c^i_z)$.

The set of 3D cones $\mathbf{CD}_{3D}$ allows us to define the *object score* as a square matrix $\Sigma$ of size $N \times N$, where $N$ is the number of objects detected by the Object Detector Transformer. The object score matrix represents whether an object is in the visual cone of each person and how close it is to their predicted gaze vector (see Fig. 3d). Each row represents an object where the rows of objects not classified as heads are zero. For rows of *head* objects, the score for each other object is equivalent to the value of the gaze cone picked at the center coordinates of the object. When no object is in the gaze cone, the corresponding row becomes zero, and then we exploit the *no cone-object* skip to compute the gaze heatmap. The *object score* matrix $\Sigma$ is used by $\mathcal{GOT}$ as an additive bias in the attention module. The rationale behind the score matrix $\Sigma$ is to exploit the strong prior coming from the gaze vector and constrain the network to focus on relevant objects in the scene.

### 3.3. Gaze Object Transformer

Although the information from the predicted gaze vector, cone, and $\Sigma$ provides important knowledge for the task at hand, accurately predicting the gaze direction is fundamentally a hard problem since the precise angle and magnitude of the vector are highly sensitive and might even introduce noise for the training procedure (see Sec. 4.4 for empirical justification). Eventually, an accurate vector estimation requires considering eye position and sight. However, such elements potentially introduce the need to use auxiliary networks, increasing the computational complexity of the overall architecture. Instead, our proposal is much simpler but effective as it does not use the gaze vector to predict the final heatmap, but we further process the output of $\mathcal{D}$ with the aid of the *object score* matrix $\Sigma$ in $\mathcal{GOT}$, which follows the same design principle as the decoder $\mathcal{D}$.

First, a stack of MHSA and FFN layers encodes a set of learnable embeddings $\mathbf{e}_g \in \mathbb{R}^{N \times C'_b}$, where $N$ is the number of predicted objects. Unlike the object detector transformer's encoder, the multi-head self-attention includes an additive bias, *i.e.* our *object score* matrix $\Sigma$. Therefore, the new attention is defined as:

$$\text{BiasedAttention}(Q, K, V) = \text{softmax}\Big(\frac{QK^T + \Sigma}{\sqrt{d_k}}\Big)V \quad (3)$$

Additionally, we mask the learnable embeddings corresponding to objects not classified as heads. The masked features of the self-attention of $\mathcal{GOT}$ are the inputs to the cross-attention module. Likewise self-attention, the cross-attention module exploits the *object score* matrix as additive bias and performs binary masking on heads for $Q$ and other objects for $K$ and $V$. We also exclude the objects with low confidence prediction or that are classified as *no-object* ($\emptyset$) (see Sec. 4.2 for details).

The output features of the cross-attention form the input to the *heatmap* MLP to predict the gaze heatmap for each head. However, since we cannot assume that an object is always present, a second MLP (*heatmap no-object* in Fig. 2) predicts the heatmap from head features only when no object is inside the visual cone. The outputs of *heatmap* MLP and *heatmap no-object* MLP are fed to *a gated operator* that selects the heatmap based on the presence (or absence) of objects in the cone of each person. Finally, an additional *watch outside* MLP, only for head objects, predicts $\mathbf{p}_{out}$, the probability that the given head gaze lies outside the frame.

### 3.4. Training objective

As we perform multiple tasks simultaneously (e.g., object localization and classification, gaze vector regression, and gaze heatmap regression), our training objective is defined as a ***weighted sum*** of all tasks.

We supervise the object localization with the weighted difference of $\mathcal{L}_1$ distance and Generalized Intersection over Union (GIoU) [32] of the target box $box$ and predicted box $box_p$, respectively, formalized as $\mathcal{L}_{box} = \lambda_{l1}\|box - box_p\| - \lambda_{giou}GIoU(box, box_p)$. Object classification loss $\mathcal{L}_{cls}$ is the cross-entropy between the ground truth label and the post-softmax distribution of the predicted class.

The gaze-related tasks involve the use of three losses: (a) gaze vector loss, (b) gaze heatmap loss, and (c) gaze watch-outside loss. The gaze vector loss is formulated as the $\mathcal{L}_2$ loss between elements of the predicted and target vector such that $\mathcal{L}_{vec} = \|\mathbf{v}_g - \mathbf{v}_p\|_2$, with $\mathbf{v}_g$ being the ground truth gaze vector and $\mathbf{v}_p$ the one predicted by our method. The watch-outside loss is a binary cross-entropy loss $\mathcal{L}_{out} = -\big[out \log(\mathbf{p}_{out}) + (1 - out)\log(1 - \mathbf{p}_{out})\big]$, where $out$ is the ground truth binary annotation of whether the person is watching outside and $\mathbf{p}_{out}$ is the predicted value. Lastly, the gaze heatmap loss is an $\mathcal{L}_2$ loss between target and predicted heatmap: $\mathcal{L}_{heat} = \lambda_{heat}\|\mathbf{H} - \mathbf{H}_p\|_2$.

| Method | Modalities | Multiperson Gaze | GazeFollow [30] | | | | | | VideoAttentionTarget [7] | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Distance ↓ | | | | *In frame* | | | | *Out of frame* | |
| | | | AUC ↑ | | Avg. | | Min. | | AUC ↑ | | Dist. ↓ | | AP ↑ | |
| | | | *Head GT* | *Real*$^\dagger$ | *Head GT* | *Real*$^\dagger$ | *Head GT* | *Real*$^\dagger$ | *Head GT* | *Real*$^\dagger$ | *Head GT* | *Real*$^\dagger$ | *Head GT* | *Real*$^\dagger$ |
| Random | | | 0.504 | 0.391 | 0.484 | 0.533 | 0.391 | 0.487 | 0.505 | 0.247 | 0.458 | 0.592 | 0.621 | 0.349 |
| Center | | | 0.633 | 0.446 | 0.313 | 0.495 | 0.230 | 0.371 | - | - | - | - | - | - |
| Fixed bias | | | - | - | - | - | - | - | 0.728 | - | 0.326 | - | 0.624 | - |
| Recasens et al. [30] | R | ✗ | 0.878 | 0.804 | 0.190 | 0.233 | 0.113 | 0.124 | - | - | - | - | - | - |
| Chong et al. [6] | R | ✗ | 0.896 | 0.807 | 0.187 | 0.207 | 0.112 | 0.120 | 0.830 | 0.791 | 0.193 | 0.214 | 0.705 | 0.651 |
| Lian et al. [23] | R | ✗ | 0.906 | 0.881 | 0.145 | 0.153 | 0.081 | 0.087 | 0.837 | 0.784 | 0.165 | 0.172 | - | - |
| Chong et al. [7] | R + T | ✗ | 0.921 | 0.902 | 0.137 | 0.142 | 0.077 | 0.082 | 0.860 | 0.812 | 0.134 | 0.146 | 0.853 | 0.849 |
| Fang et al. [13] | R + D | ✗ | 0.922 | - | 0,124 | - | 0.067 | - | 0.905 | - | 0.108 | - | 0.896 | - |
| Bao et al. [2] | R + D + P | ✗ | 0.928 | - | 0.122 | - | - | - | 0.885 | - | 0.120 | - | 0.869 | - |
| Jin et al. [19] | R + D | ✗ | 0.920 | - | 0.118 | - | 0.063 | - | 0.900 | - | 0.104 | - | 0.895 | - |
| Tonini et al. [34] | R + D | ✗ | 0.927 | 0.894 | 0.141 | 0.165 | - | - | 0.940 | 0.894 | 0.129 | 0.182 | - | - |
| Qiaomu et al. [26] | R + D + T | ✗ | 0.934 | - | 0.123 | - | 0.065 | - | 0.917 | - | 0.109 | - | 0.908 | - |
| Tu et al. [35] | R | ✓ | - | 0.917 | - | 0.133 | - | 0.069 | - | 0.904 | - | 0.126 | - | 0.854 |
| Tu et al. [35]$^\star$ | R | ✓ | - | 0.915 | - | 0.104 | - | 0.055 | - | 0.891 | - | 0.229 | - | 0.809 |
| Our method | R | ✓ | - | **0.922** | - | 0.072 | - | 0.033 | - | 0.923 | - | **0.102** | - | **0.944** |
| Our method | R + D | ✓ | - | **0.922** | - | **0.069** | - | **0.029** | - | **0.933** | - | 0.104 | - | 0.934 |

Table 2: Evaluation on the GazeFollow [30] and VideoAttentionTarget [7] datasets. *Head GT* refers to using carefully labeled ground-truth head crops and head locations in training and testing. *Real* indicated with $^\dagger$ is the implementation of [35], which applies an additional SOTA head detection network to predict the head location for real-world applications. We produce only [34]'s *Real* results (see text for details). $^\star$ indicates our implementation. $R$, $D$, $T$, and $P$ stand for RGB, depth, temporal processing, and 2D-pose, respectively. Refer to Supp. Mat. for Angular Error results.

# 4. Experiments

## 4.1. Datasets and Evaluation metrics

**Datasets.** Our model is trained and tested on both GazeFollow [30] and VideoAttentionTarget [7] datasets. **GazeFollow** [30] is a large-scale *image* dataset containing over 122K images in total with more than 130K people. The test images include gaze and head location annotations performed by up to 10 people for a single person in the scene while the training set contains only one annotator's judgment indicating gaze and head locations. **VideoAttentionTarget** [7] is composed of YouTube *video* clips, each has a length of up to 80 seconds. It includes 109574 in-frame and 54967 out-of-frame gaze annotations together with the head locations. Both the training and test sets contain one gaze annotation per person. Given that we do not use the *temporal* information in our model, we randomly select one image for every 5 consecutive frames, allowing us to avoid overfitting. This setup is the same with SOTA [2, 13, 19, 34, 35].

**Evaluation Metrics.** We evaluate the performance of the proposed method in terms of **gaze target detection** and **object class detection and localization**. For the former task, we use all standard metrics [6, 7] described as follows. **AUC** assesses the confidence of the predicted gaze heatmap *w.r.t.* the gaze ground-truth. **Distance** (Dist.) is the $\mathcal{L}_2$ between the ground-truth gaze point and the predicted gaze location, which is the point with the maximum confidence on the gaze heatmap. **Angular Error** (Ang. Err.) is the angle between predicted and ground-truth gaze vector. In GazeFollow, it is a standard to declare both the minimum and average distances. **I/O gaze AP** is the average precision used to evaluate the *out-of-frame* probability of the gaze in VideoAttentionTarget. We use the standard metric **Mean Average Precision (mAP)** for object class detection and localization. In that case, a prediction is correct if the class label of the predicted bounding box and the ground truth bounding box are the same and the Intersection over Union ($IoU$) between them is greater than a $threshold$ value.

## 4.2. Implementation details

$\mathcal{B}$ is a ResNet-50 pretrained on ImageNet [9] and the Object Detector Transformer follows the DETR [40] architecture. We train all our components (Object Detector Transformer, $\mathcal{GOT}$, *Gaze Cone Predictor*, and MLPs) with Adam optimizer and a learning rate of $1 \times 10^{-4}$ for 80 epochs, then we drop the learning rate by 10 times and train for 20 epochs. Differently, $\mathcal{B}$ has a learning rate 10 times smaller, i.e. $1 \times 10^{-5}$. Furthermore, we perform matching between predictions and ground-truth samples as described in [35]. The FoV angle of the cone predictor is set to 120°, corresponding to the binocular FoV of humans [17]. $\mathcal{GOT}$ keeps only queries of objects classified as heads and with confidence above $0.5$. Conversely, the keys and values are those of objects (heads included) with the confidence above $0.5$, which are not classified as *no-object*. The final loss is the weighted sum of the defined objectives. We set $\lambda_{gious} = 2.5$ and $\lambda_{heat} = 2$. The other losses are summed up without any weighting. We use a SoTA monocular depth estimator [28] to obtain depth maps corresponding to each scene image. Note that we use depth information only for gaze cone building without learning additional depth features. More details are available in Supp. Mat.

### 4.3. Comparison with State-of-the-Art

The gaze target detection performance of our method is compared with the SOTA in Table 2. Recalling that the cropped head images and the head locations are required for traditional methods (i.e., SOTA except [35]) and these methods are evaluated when the ground-truth head locations are granted (referred to as "Head GT"), we proceed with the evaluation procedure of [35], summarized as follows. Tu et al. [35] employ additional head detectors to automatically obtain the heads position, which is given to the traditional models, providing their real-world application performance. We inherit the corresponding results from [35] and refer to them as "Real". For the methods whose "Real" results are not provided by [35], we obtain the results using RetinaFace [10] to detect heads position. However, we are able to perform this only for the method whose code is publicly available: [34].

As we can see from the results, our method only with RGB data outperforms existing SOTA on all datasets for all metrics. Such a performance is important to emphasize since several SOTA perform relatively poorly even though they use multi-modalities [13, 19] or temporal data [7]. Particularly, for VideoAttentionTarget [7] dataset, our method achieves better scores compared to many complex methods relying on several pretrained task-specific backbones (e.g., 2D-pose estimation) [2] or leveraging the temporal dimensionality of the data [26] while both utilize RGB and depth maps. Our better performance *w.r.t.* Transformer-based [35] is also conspicuous. Furthermore, when both RGB and depth are taken into account, our method performance slightly improves on average. Recalling that we use depth information only during gaze cone production without requiring additional (pretrained) CNN to learn depth features as in [19, 34] or needing to detect the eyes as in [13], the corresponding results are momentous. Particularly, our minimum and average distance and mAP results are always the best whether or not others were evaluated within "Head GT" or "Real" settings. This shows that the proposed method is notably good at predicting if the gaze is located inside or outside the frame, the gaze heatmaps, and eventually, a single pixel gaze point that our model predicts per person is much closer to the ground truth-gaze point.

### 4.4. Ablation Study

The ablation study is performed on both GazeFollow [30] and VideoAttentionTarget [7] datasets, whose results can be found in Table 3.

**Gaze Object Transformer.** If we do not use $\mathcal{GOT}$, it is still possible to predict the gaze heatmap using the features of $\mathcal{D}$. As seen from the results (first row of the ablations for each dataset), $\mathcal{D}$ features alone are insufficient to reach SOTA results for gaze target detection. Whereas including $\mathcal{GOT}$ boosts the results for all metrics and datasets (second row of the ablations for each dataset).

| $\mathcal{GOT}$ | OM | GC | NOCS | GazeFollow [30] | | |
|---|---|---|---|---|---|---|
| | | | | AUC ↑ | Avg. dist. ↓ | Min. dist. ↓ |
| ✗ | ✗ | ✗ | ✗ | 0.864 | 0.110 | 0.061 |
| ✓ | ✗ | ✗ | ✗ | 0.918 | 0.075 | 0.038 |
| ✓ | ✓ | ✗ | ✗ | 0.919 | 0.073 | 0.033 |
| ✓ | ✓ | ✓ | ✗ | 0.905 | 0.090 | 0.051 |
| ✓ | ✓ | ✓ | ✓ | **0.922** | **0.072** | **0.033** |

| $\mathcal{GOT}$ | OM | GC | NOCS | VideoAttentionTarget [7] | | |
|---|---|---|---|---|---|---|
| | | | | AUC ↑ | Dist. ↓ | AP ↑ |
| ✗ | ✗ | ✗ | ✗ | 0.811 | 0.271 | 0.77 |
| ✓ | ✗ | ✗ | ✗ | 0.902 | 0.125 | 0.92 |
| ✓ | ✓ | ✗ | ✗ | 0.907 | 0.112 | **0.94** |
| ✓ | ✓ | ✓ | ✗ | 0.909 | 0.154 | **0.94** |
| ✓ | ✓ | ✓ | ✓ | **0.923** | **0.101** | **0.94** |

Table 3: Ablation study on GazeFollow [30] and VideoAttentionTarget [7]. $OM$, $GC$, $NCOS$ stand for object masking, gaze cone, and *no cone-obj* skip, respectively.

**Object Masking.** By definition, Transformer attention attends to every token in a sequence and tries to learn relationships between all elements. In our case, this refers to computing the interaction between every object. Instead, our design retains only the *queries* to be of those elements recognized as heads and *keys* and *values* to be those of any other object/head. In this way, we obtain an improvement across both datasets for all the metrics (third row of the ablations). Furthermore, we obtain an interpretable attention matrix of interaction between heads and objects.

**Gaze Cone and No cone-object Skip.** Gaze cone building assigns a score into $\Sigma$ inversely proportional to the distance from the gaze vector for the objects inside the cone. This acts similarly to a temperature to skew the softmax operation inside the attention towards the objects more probable to be looked at. As seen, the gaze cone alone might not be sufficient to improve the performance of the method (fourth row of the table). We attribute this to the cases where we cannot find a meaningful object inside the gaze cone, meaning that the $\Sigma$ row corresponding to the face is empty, and attention does not operate on any feature, hindering the performance of the *heatmap* MLP. To solve this, we design a *no cone-object* skip, which allows building a heatmap starting from $\mathcal{D}$ features. In such cases, a gating mechanism allows selecting which heatmap to use depending on the presence of objects in the cone. When we use this mechanism in conjunction with cone building (fifth row of the table, aka *full proposed method*), we obtain the best results consistently across the datasets, proving the effectiveness of focusing on relevant objects in the scene.

### 4.5. Gazed-object class detection and localization

This section reports the evaluations regarding gazed-object class detection and localization performance. Our

| Method | # params. ↓ | AP ↑ | $AP_{50}$ ↑ | $AP_{75}$ ↑ |
|---|---|---|---|---|
| Tu et al. [35] | 43M | 0.01 | 0.03 | 0.01 |
| Tu et al. [35] + DETR [40] | 84M | 0.04 | 0.12 | 0.02 |
| Ours + DETR [40] | 97M | 0.03 | 0.10 | 0.01 |
| Ours | **54M** | **0.14** | **0.22** | **0.15** |

Table 4: Gazed-object classification and localization performance. The computational complexity is reported in terms of parameters.

method is notably different from using an auxiliary object detector accompanying a gaze target detection model. Still, in order to empirically highlight the difference, we combine the model of Tu et al. [35] with DETR [40] pretrained on COCO [24]. To this end, given a produced gaze heatmap of [35], we use the bounding box proposal of DETR which contains the highest value of the heatmap (notice that this is in line with ground-truth gaze heatmap construction [7]). Similarly, we also combined the results of DETR with our model's gaze heatmap predictions. Moreover, we include [35] in the comparisons by determining a bounding box that surrounds the gaze heatmaps of [35]. In that case, $AP$ was calculated only for object locations, discarding the object class prediction. The corresponding results given in Table 4 were performed using the COCO-subset of the GazeFollow dataset [30] providing the ground-truth object class and location information. Notice that our model was not particularly trained on COCO ground-truth object classes and locations but was trained on the full set of the GazeFollow and its *gaze* annotations. Instead, DETR was trained on the full COCO dataset [24]. That setting should rather be advantageous for DETR since it is aware of all object classes. Overall, the results show the relative effectiveness of our model for gazed-object prediction while it is also the most efficient in terms of the number of parameters.

| | 10% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|---|---|
| Ours 2D | 0.980 | 0.977 | 0.973 | 0.970 | 0.964 | 0.961 | 0.958 | 0.954 | 0.943 | 0.922 |
| Ours 3D | 0.980 | 0.977 | 0.972 | 0.967 | 0.961 | 0.957 | 0.953 | 0.950 | 0.944 | 0.922 |
| [35] | 0.973 | 0.967 | 0.964 | 0.957 | 0.953 | 0.948 | 0.945 | 0.940 | 0.932 | 0.915 |

Table 5: AUC between our 2D and 3D method and [35] w.r.t. increasing variance levels.

### 4.6. The effect of variance in gaze annotations

We compare the AUC of the proposed method with [35] considering the multiple annotations that the GazeFollow dataset's test split provides. In some cases, the annotators' consensus is low, as highlighted in [26], which motivated us to evaluate the methods under different levels of variance across the individual gaze annotations. The calculation of the annotation variance and extensive discussions are given in Supp. Mat. The results presented in Table 5 show the permanent better performance of our model both in 2D and 3D *w.r.t.* [35] while, as expected, with lower variance all methods perform better. We speculate that the lower performance of Ours-3D *w.r.t.* Ours-2D can be since the human annotations were collected on 2D images.

### 4.7. Qualitative Results

We visualize gaze heatmaps of our method and [35] in Fig. 4 on the GazeFollow dataset. Our predictions are more accurate compared to [35] in line with the quantitative results. Refer to Supp. Mat. for more qualitative results, including some less accurate performance of the proposed method *w.r.t.* the ground-truth.

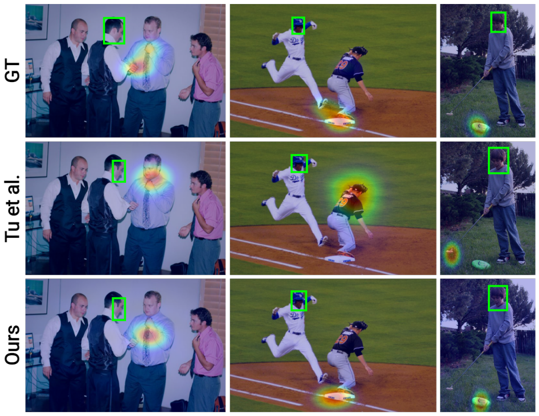


Figure 4: Qualitative results of our method (bottom) and Tu et al. [35] (middle) *w.r.t.* the ground-truth (top). For simplicity, we show only one person's gaze.

## 5. Discussion & Conclusion

We have presented a new end-to-end Transformer-based gaze target detector simultaneously predicting the *object class* and the *location* of the gazed-object. The latter is advantageous *w.r.t.* prior art as it improves explainability. Extensive experiments validate our approach's better performance for gaze behavior understanding, promising its usefulness in real-world human interaction analysis.

**Broader Impacts.** We target a human-centric task and consequently, our model, in some cases, might need to process human faces. This might result in issues regarding privacy protection, therefore policy review should be considered when using this model in real-world applications.

**Limitations & Future Work.** As expected from a Transformer-based model, our network also has slow convergence, requiring long training epochs. Future work will investigate gaze-target detection within the open-set object detection paradigms.

**Acknowledgments.** This work was supported by the EU H2020 SPRING (No. 871245) and AI4Media (No. 951911) projects, and the MUR PNRR project FAIR - Future AI Research (PE00000013) funded by the NextGenerationEU. It was carried out in the Vision and Learning joint laboratory of FBK and UNITN.

## References

[1] Henny Admoni and Brian Scassellati. Social eye gaze in human-robot interaction: a review. *Journal of Human-Robot Interaction*, 6(1):25–63, 2017. 1

[2] Jun Bao, Buyu Liu, and Jun Yu. Escnet: Gaze target detection with the understanding of 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14126–14135, 2022. 1, 2, 3, 6, 7, 12

[3] Francesca Capozzi, Cigdem Beyan, Antonio Pierro, Atesh Koul, Vittorio Murino, Stefano Livi, Andrew P Bayliss, Jelena Ristic, and Cristina Becchio. Tracking the leader: Gaze behavior in group interactions. *Iscience*, 16:242–249, 2019. 1

[4] Yihua Cheng, Feng Lu, and Xucong Zhang. Appearance-based gaze estimation via evaluation-guided asymmetric regression. In *Proc. of ECCV*, September 2018. 1

[5] Dae-Yong Cho and Min-Koo Kang. Human gaze-aware attentive object detection for ambient intelligence. *Engineering Applications of Artificial Intelligence*, 106:104471, 2021. 2

[6] Eunji Chong, Nataniel Ruiz, Yongxin Wang, Yun Zhang, Agata Rozga, and James M. Rehg. Connecting gaze, scene, and attention: Generalized attention estimation via joint modeling of gaze and scene saliency. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1, 2, 6

[7] 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. 1, 2, 3, 6, 7, 8, 11, 12, 14

[8] Kim M Dalton, Brendon M Nacewicz, Tom Johnstone, Hillary S Schaefer, Morton Ann Gernsbacher, Hill H Goldsmith, Andrew L Alexander, and Richard J Davidson. Gaze fixation and the neural circuitry of face processing in autism. *Nature neuroscience*, 8(4):519–526, 2005. 1

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 6

[10] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 7, 12

[11] S Gareth Edwards, Lisa J Stephenson, Mario Dalmaso, and Andrew P Bayliss. Social orienting in gaze leading: a mechanism for shared attention. *Proceedings of the Royal Society B: Biological Sciences*, 282(1812):20151141, 2015. 1

[12] Nathan J Emery. The eyes have it: the neuroethology, function and evolution of social gaze. *Neuroscience & biobehavioral reviews*, 24(6):581–604, 2000. 1

[13] Yi Fang, Jiapeng Tang, Wang Shen, Wei Shen, Xiao Gu, Li Song, and Guangtao Zhai. Dual attention guided gaze target detection in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11390–11399, 2021. 1, 2, 3, 6, 7, 12

[14] Kenneth Alberto Funes Mora, Florent Monay, and Jean-Marc Odobez. Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras. In *Proceedings of the symposium on eye tracking research and applications*, pages 255–258, 2014. 1

[15] Zidong Guo, Zejian Yuan, Chong Zhang, Wanchao Chi, Yonggen Ling, and Shenghao Zhang. Domain adaptation gaze estimation by embedding with prediction consistency. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 1

[16] Anshul Gupta, Samy Tafasca, and Jean-Marc 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*, pages 5041–5050, 2022. 4

[17] Ian P Howard, Brian J Rogers, et al. *Binocular vision and stereopsis*. Oxford University Press, USA, 1995. 4, 6

[18] Zhengxi Hu, Dingye Yang, Shilei Cheng, Lei Zhou, Shichao Wu, and Jingtai Liu. We know where they are looking at from the rgb-d camera: Gaze following in 3d. *IEEE Transactions on Instrumentation and Measurement*, 71:1–14, 2022. 1

[19] Tianlei Jin, Qizhi Yu, Shiqiang Zhu, Zheyuan Lin, Jie Ren, Yuanhai Zhou, and Wei Song. Depth-aware gaze-following via auxiliary networks for robotics. *Engineering Applications of Artificial Intelligence*, 113:104924, 2022. 1, 2, 3, 6, 7, 12

[20] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics, 2023. 12

[21] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6912–6921, 2019. 1

[22] Yin Li, Miao Liu, and Jame Rehg. In the eye of the beholder: Gaze and actions in first person video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[23] Dongze Lian, Zehao Yu, and Shenghua Gao. Believe it or not, we know what you are looking at! In *Asian Conference on Computer Vision*, pages 35–50. Springer, 2018. 1, 2, 6, 12

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 8

[25] Michele Mazzamuto, Francesco Ragusa, Antonino Furnari, Giovanni Signorello, and Giovanni Maria Farinella. Weakly supervised attended object detection using gaze data as annotations. In *Image Analysis and Processing–ICIAP 2022: 21st International Conference, Lecce, Italy, May 23–27, 2022, Proceedings, Part II*, pages 263–274. Springer, 2022. 2

[26] 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. 1, 2, 3, 6, 7, 8, 11

[27] Borna Noureddin, Peter D Lawrence, and CF Man. A non-contact device for tracking gaze in a human computer interface. *Computer Vision and Image Understanding*, 98(1):52–82, 2005. 1

[28] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020. 6, 12

[29] Keith Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372, 1998. 1

[30] Adria Recasens, Aditya Khosla, Carl Vondrick, and Antonio Torralba. Where are they looking? In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 1, 2, 6, 7, 8, 11, 12, 13, 15, 16

[31] Adrià Recasens, Carl Vondrick, Aditya Khosla, and Antonio Torralba. Following gaze in video. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1444–1452, 2017. 1, 2

[32] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 5

[33] Boris Schauerte and Rainer Stiefelhagen. "look at this!" learning to guide visual saliency in human-robot interaction. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 995–1002, 2014. 2

[34] Francesco Tonini, Cigdem Beyan, and Elisa Ricci. Multimodal across domains gaze target detection. In *Proceedings of the 2022 International Conference on Multimodal Interaction*, pages 420–431, 2022. 1, 2, 3, 6, 7, 12

[35] Danyang Tu, Xiongkuo Min, Huiyu Duan, Guodong Guo, Guangtao Zhai, and Wei Shen. End-to-end human-gaze-target detection with transformers. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2192–2200. IEEE, 2022. 1, 2, 3, 6, 7, 8, 11, 12, 13, 14, 15

[36] Daniel Weber, Wolfgang Fuhl, Andreas Zell, and Enkelejda Kasneci. Gaze-based object detection in the wild. In *2022 Sixth IEEE International Conference on Robotic Computing (IRC)*, pages 62–66, 2022. 2

[37] Lijun Yin and Michael Reale. Real time eye tracking for human computer interaction, Nov. 11 2014. US Patent 8,885,882. 1

[38] Akishige Yuguchi, Tomoaki Inoue, Gustavo Alfonso Garcia Ricardez, Ming Ding, Jun Takamatsu, and Tsukasa Ogasawara. Real-time gazed object identification with a variable point of view using a mobile service robot. In *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1–6. IEEE, 2019. 2

[39] Kiwon Yun, Yifan Peng, Dimitris Samaras, Gregory J Zelinsky, and Tamara L Berg. Exploring the role of gaze behavior and object detection in scene understanding. *Frontiers in psychology*, 4:917, 2013. 2

[40] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *9th International Conference on Learning Representations, ICLR 2021*, 2021. 6, 8, 12

## A. The effect of variance in gaze annotations

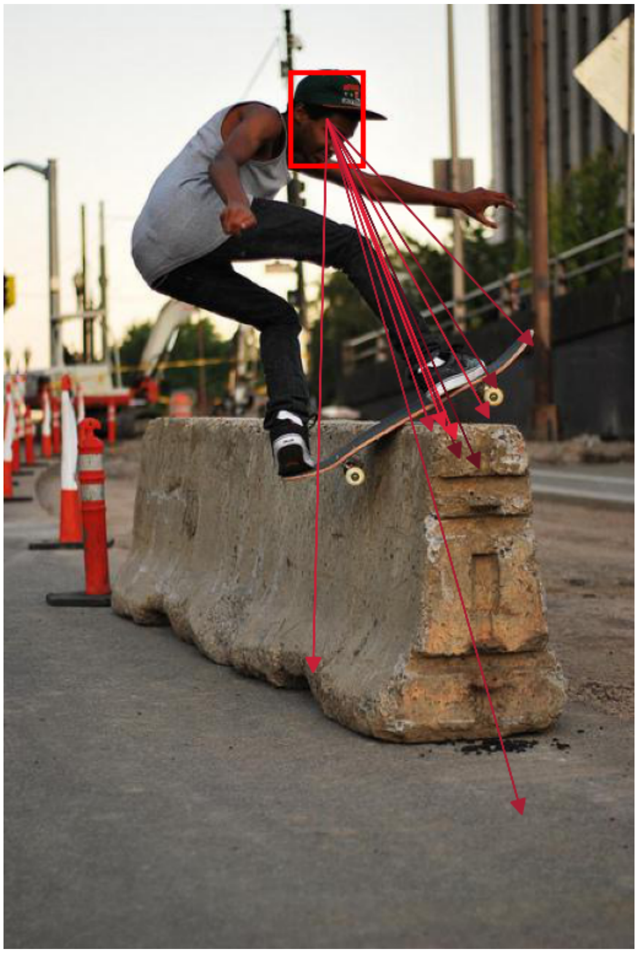


Figure 5: An image from the GazeFollow dataset's [30] test split which has several annotations for the gaze point with high annotation variance.

As mentioned in the paper, the GazeFollow dataset [30] contains a single gaze point annotation for a single person in a scene in its training split. However, its test splits include several numbers of annotations with respect to a single person's gaze. The number of annotations can be varying up to 10 different gaze points for each person. Such an annotation procedure would not present an issue if all the annotators reached a consensus regarding the gaze point, however, as also shown in [26] the test split annotations of that dataset per person can vary remarkably. An example image and the corresponding gaze points demonstrating the variety across the annotations are given in Fig. 5.

On the other hand, the standard metric used to make evaluations on this dataset, aka AUC does not consider the (possible) varieties across the annotations. For this reason, herein as well as in the main paper, we present an additional evaluation procedure, which considers the multiple annotations that GazeFollow test split provides (see Fig. 6). That evaluation procedure can be described as follows. For each gaze point annotation corresponding to a single person, we compute its distance to the corresponding average gaze annotation point. We record all the distances for the whole test split and compose a distribution from them. Then, given this distribution, we keep the gaze points falling inside a certain threshold (shown as variance retained in the figure), and we opt for the deciles for easiness of computation. For each decile, we compute the AUC and report it in Fig. 6. As seen, our method is extremely effective when there is high annotation consensus, *i.e.* the distance from the average point falls in the first decile (*i.e.*, 0-10% variance retained in the figure); the performance slightly decreases until the eighth decile (*i.e.*, 70-80% variance retained in the figure), with the last 20% representing high noise annotations (*i.e.*, 80-100% variance retained in the figure) where the performance lowers at a faster rate. When we compare our performance against the state-of-the-art method of [35], one can observe a consistently higher performance for all cases both in 2D and 3D versions of our method. We speculate that the lower performance of Ours-3D *w.r.t.* Ours-2D can be since the human annotations were collected on 2D images.

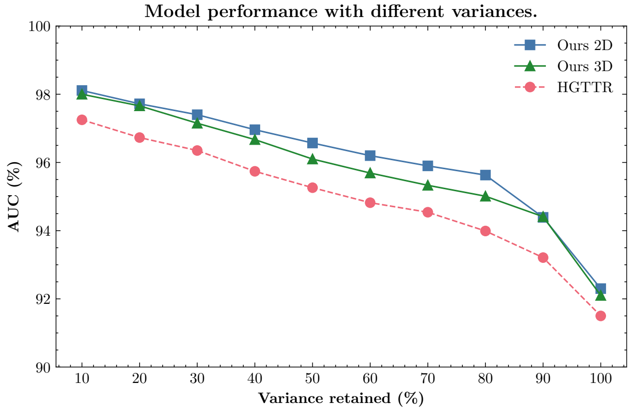


Figure 6: Performance of Our model (2D and 3D) and Tu et al. [35] *w.r.t.* the variance in the ground-truth annotations of the GazeFollow dataset [30].

## B. Additional evaluation on GazeFollow [30] and VideoAttentionTarget [7]

Table 6 reports the *Angular Error* [30] (i.e. the angle between predicted and ground-truth gaze vector) results and compare it with SOTA. Our method produces the best results out of all, while Ours (3D) is better than Ours (2D).

|  | Ours (2D) | Ours (3D) | [35]* | [34]* | [19] | [2] | [13] | [7]* | [23] | [30] |
|---|---|---|---|---|---|---|---|---|---|---|
| Min. ↓ | 4.0° | **3.5**° (−12.5%) | 6.6° | 8.1° | — | — | — | 9.1° | 8.8° | — |
| Avg. ↓ | 7.7° | **7.2**° (−6.2%) | 11.0° | 19.5° | 14.8° | 14.6° | 14.9° | 20.5° | 17.6° | 24.0° |
| Max. ↓ | 20.1° | **19.3**° (−3.9%) | 22.5° | 37.0° | — | — | — | 37.9° | — | — |

Table 6: Angular error on GazeFollow [30] ⋆ means our implementation. Improvements are w.r.t. "Ours (2D)".

## C. Implementation Details

We implemented our method in PyTorch and relied on the official code of DETR [40] as the backbone. The heads of DETR [40], *i.e.* the two MLPs for object classification and detection, were replaced by two larger MLPs that allow us to predict the location and classification of objects in the scene including the *heads*. Therefore, the number of classes of objects is adapted to accommodate the *head* class. We used a SOTA object detector, YOLOv8 [20], to pseudo-annotate objects in images that lack object annotations. This has been needed since the used datasets (except the COCO subset of the GazeFollow dataset) do not provide object annotations. We finetuned the *Object Detector Transformer* using head locations given in the used datasets as well as automatically obtained using an additional head detector, RetinaFace [10], and for other objects extracted from YOLOv8. RetinaFace was necessary (but other head detectors can be also adapted as shown in Tu et al. [35]) as we observed that both Tu et al. [35], and our method could not converge without head annotations of all heads in the image. The depth images were obtained by processing both datasets with a SOTA monocular depth estimation method called MIDAS [28]. The source code of our proposed method can be found as `src.zip` archive for additional details. Due to limitations in the total size of the supplementary materials, we will publicly provide checkpoints for both datasets upon acceptance.

## D. Qualitative Results

In this section, we provide additional qualitative results of the gaze heatmaps and the head bounding box of the gaze source (*i.e.*, a person's head) and demonstrate the improved performance of our method *w.r.t.* the current state-of-the-art (SOTA) for both GazeFollow [30] and VideoAttentionTarget [7] datasets. Furthermore, we discuss some example cases in which our method has relatively lower performance ($AUC < 70\%$) *w.r.t.* ground-truth as well as Tu et al. [35]. Lastly, we compare our methods' versions in 2D and 3D and demonstrate the latter's effectiveness in challenging scenarios.

**Comparison with SOTA and ground-truth.** Fig. 7 and Fig. 8 compare our predictions with respect to the ground truth and the predictions of Tu et al. [35] on both datasets, GazeFollow [30] and VideoAttentionTarget [7]. As we can see, our model precisely predicts the gaze in many scenes where [35] is not able to. More importantly, we can see that predictions of both our method and Tu et al. [35] are in the field of view of the person whose gaze is to be predicted. However, [35] favors image regions closer to the gaze source (*i.e.* person's head).

**Relatively low-performing predictions.** Fig. 9 presents example images in which our method relatively performs worse. Notice that such images are highly challenging and most of the time also SOTA [35] underperform, *e.g.* in the second row of Fig. 9, where the head-pose makes it difficult to accurately predict the gaze. Conversely, when the face is not fully visible, e.g., in the third and fifth row of Fig. 9, we predict scattered heatmaps that cover the gaze point.

**The contribution of 3D gaze cone.** Fig. 10 demonstrates the results of our method with the 2D or 3D gaze cone (see main paper for additional details). This comparison aims to highlight the importance of the 3D cone particularly in challenging scenes. As the quantitative results in the main paper showed, the advantage of the 3D cone is especially visible in terms of the average and minimum distance between the ground-truth gaze point and the point of maximum confidence of the gaze heatmap. The example images in Fig. 10 demonstrate that in complex scenes, the 3D gaze vector and the corresponding 3D gaze cone help to decipher which object the person is looking at. Moreover, when the predictions with 2D cone are already high (*e.g.* the first row of Fig. 10), the 3D cone counterpart further consolidates the center of the heatmap towards the object, resulting in better performance.

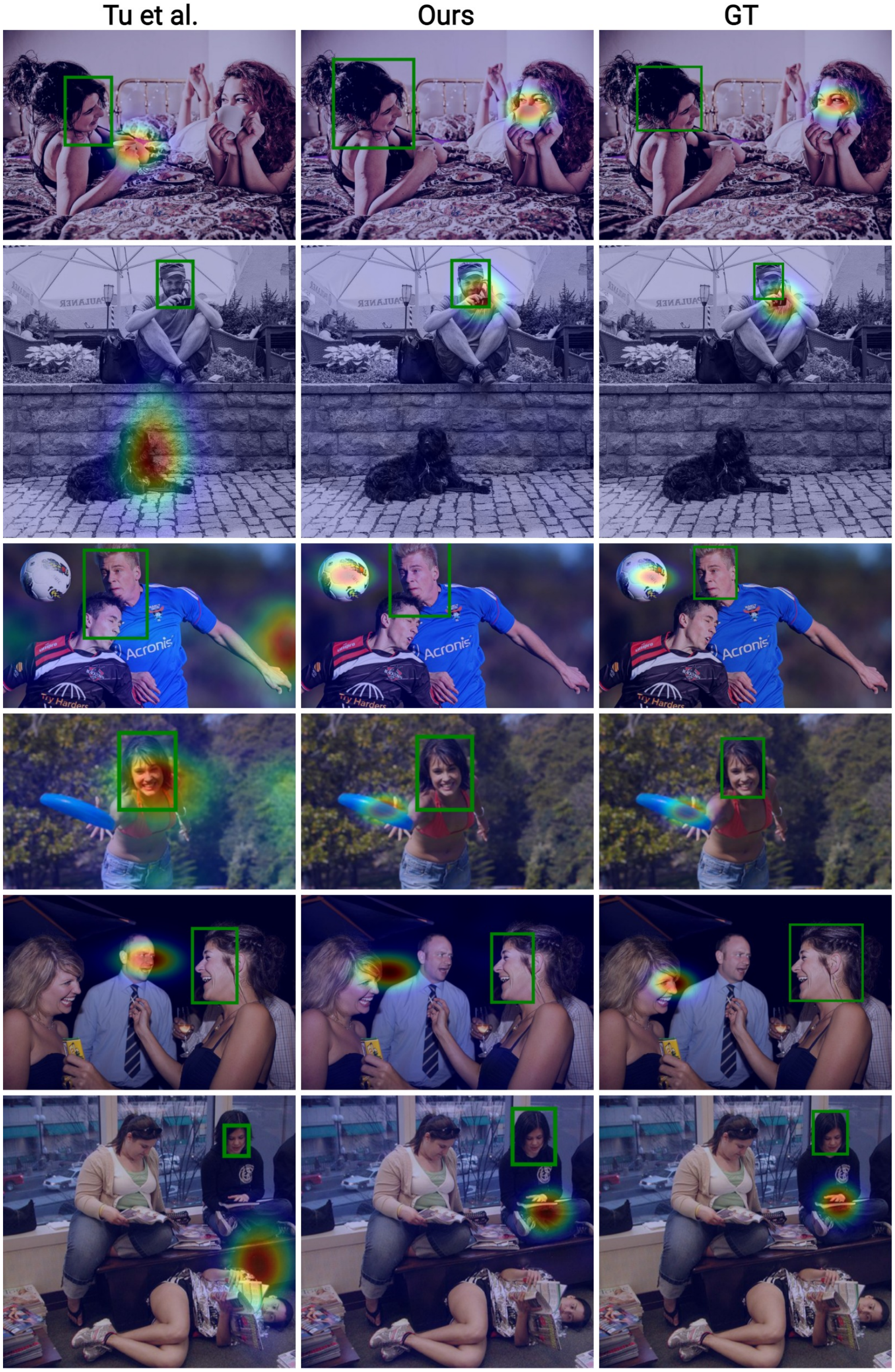


Figure 7: Qualitative results of our method (center) and Tu et al. [35] (left) *w.r.t.* the ground-truth annotations of the GazeFollow [30] dataset (right). Please note that we only show gaze predictions for the person whose gaze is included in the ground truth. The green boxes show the person-in-interest for the ground truth while they are the detected head for Our and Tu et al. [35]. Even though our method can detect the gaze of multiple persons in the scene simultaneously, for better visualization, we plot the predicted heatmaps and head locations per person.

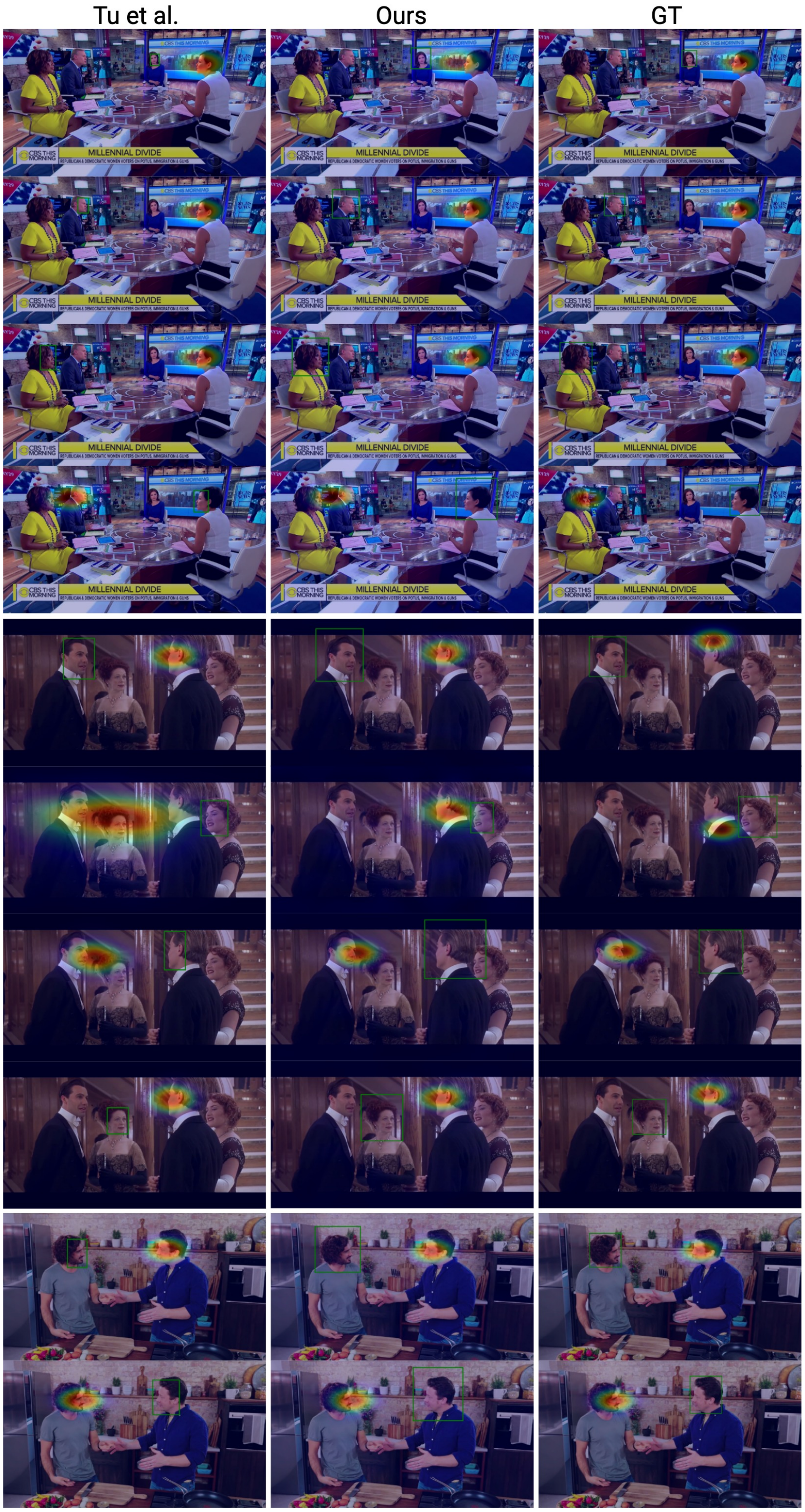


Figure 8: Qualitative results of our method (center) and Tu et al. [35] (left) *w.r.t.* the ground-truth annotations of the VideoAttentionTarget [7] dataset (right). Please note that we only show gaze predictions for the person whose gaze is included in the ground truth. The green boxes show the person-in-interest for the ground truth while they are the detected head for Our and Tu et al. [35]. Even though our method can detect the gaze of multiple persons in the scene simultaneously, for better visualization, we plot a single person's predicted heatmaps and head location.

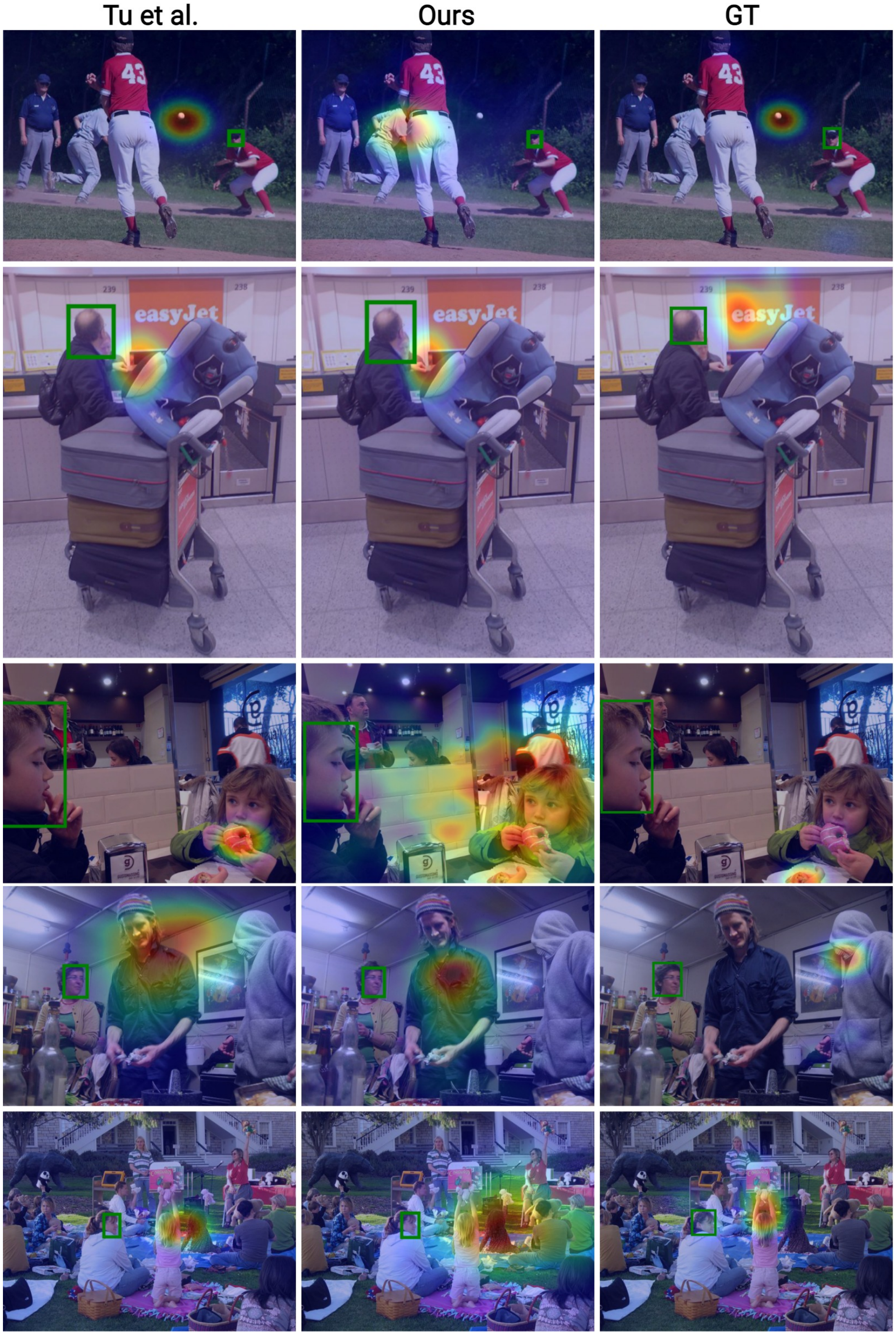


Figure 9: Qualitative results in which Our method performs relatively lower since the images are highly challenging due to several reasons (see text for details). Our method (center) and Tu et al. [35] (left) *w.r.t.* the ground-truth annotations of the GazeFollow [30] dataset (right). The green boxes show the person-in-interest for the ground truth while they are the detected head for Our and Tu et al. [35]. Even though our method can detect the gaze of multiple persons in the scene simultaneously, for better visualization, we plot a single person's predicted heatmaps and head location.

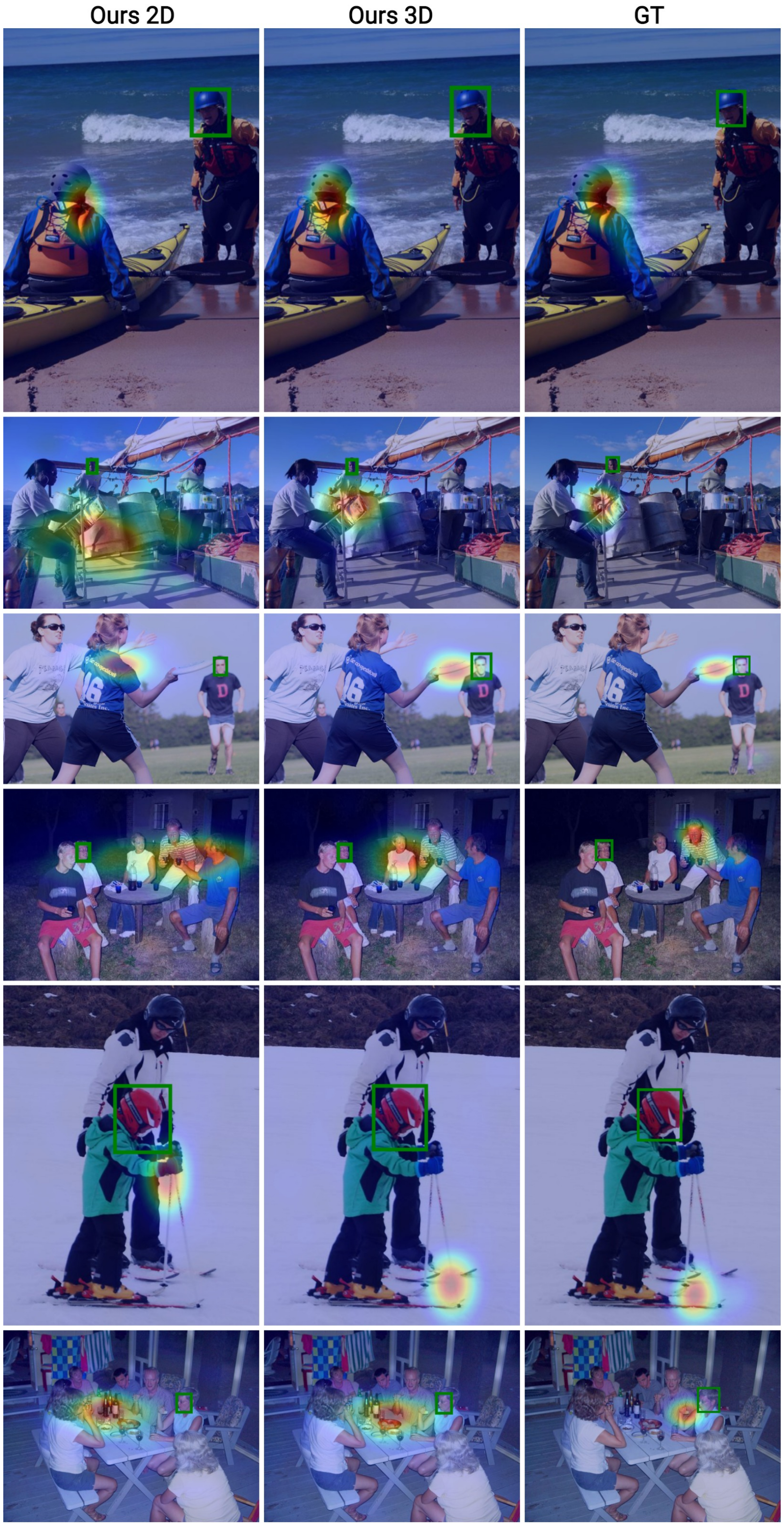


Figure 10: Qualitative results of our 2D (left) and 3D (center) method *w.r.t.* the ground-truth annotations of the GazeFollow [30] dataset (right), showing the importance of 3D-gaze cone building. Even though our method can detect the gaze of multiple persons in the scene simultaneously, for better visualization, we plot a single person's predicted heatmaps and head location.