GaTector: A Unified Framework for Gaze Object Prediction

Binglu Wang,*   Tao Hu,   Baoshan Li,   Xiaojuan Chen,   Zhijie Zhang
Xi'an University of Architecture and Technology,   China

Abstract

Gaze object prediction is a newly proposed task that aims to discover the objects being stared at by humans. It is of great application significance but still lacks a unified solution framework. An intuitive solution is to incorporate an object detection branch into an existing gaze prediction method. However, previous gaze prediction methods usually use two different networks to extract features from scene image and head image, which would lead to heavy network architecture and prevent each branch from joint optimization. In this paper, we build a novel framework named GaTector to tackle the gaze object prediction problem in a unified way. Particularly, a specific-general-specific (SGS) feature extractor is firstly proposed to utilize a shared backbone to extract general features for both scene and head images. To better consider the specificity of inputs and tasks, SGS introduces two input-specific blocks before the shared backbone and three task-specific blocks after the shared backbone. Specifically, a novel Defocus layer is designed to generate object-specific features for the object detection task without losing information or requiring extra computations. Moreover, the energy aggregation loss is introduced to guide the gaze heatmap to concentrate on the stared box. In the end, we propose a novel wUoC metric that can reveal the difference between boxes even when they share no overlapping area. Extensive experiments on the GOO dataset verify the superiority of our method in all three tracks, i.e. object detection, gaze estimation, and gaze object prediction.

1. Introduction

Gaze estimation (GE) aims at determining the direction and point that a person is staring at [30]. As gaze behavior is an essential aspect of human social behavior [7, 19], we can infer potential information based on the staring object. For example, in front of the bus station, a person looking at his watch may indicate that he has something urgent to do. Customers staring at the product may want to purchase it in the shopping mall. The stared object can generally reveal our state, *e.g.* what they are doing, or what they plan to do.
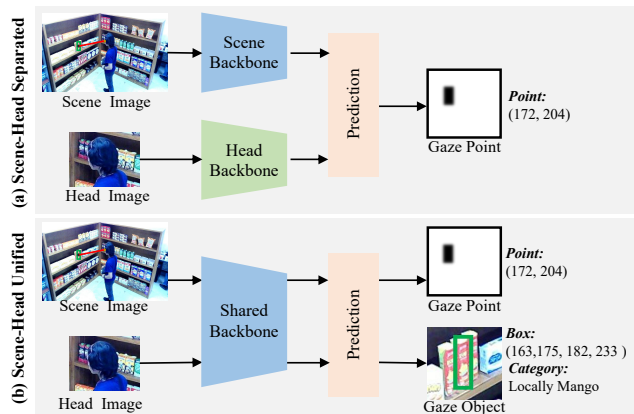
---

*Corresponding author.

Figure 1. Illustration of the gaze object prediction approaches. (a) Previous gaze prediction methods use two separate backbones to tackle scene image and head image, respectively. (b) The proposed SGS feature extractor can produce features from scene image and head image in a unified manner.

In the gaze estimation community, researchers usually employ two separated backbones to process the entire scene image and the head image. This scene-head separated structure was first proposed by Recasens *et al.* [30], where one backbone captures holistic cues from the entire scene image, and the other backbone analyzes the details from the head image. Later, Lian *et al.* [21] proposed the multi-scale gaze direction fields to analyze the head image precisely. Recently, Chong *et al.* [4] elaborately designed interactions between the head branch and the scene branch and utilized deconvolutional layers to produce a fine-grained heatmap. In general, the gaze estimation performance has kept improving in recent years while the network architecture gradually gets complex. Existing models only predict the gaze area that people may stare at rather than precisely predicting the location of the object been be stared at. Tomas *et al.* [38] recently pointed out that it has significant practical usage to identify the stared thing. As shown in Fig. 1, from the bounding box of the stared item, we can infer that the person is likely to buy the *Locally Mango* product, which is beyond the scope of the traditional gaze estimation task.

However, performing gaze object prediction is nontrivial and faces the following challenges, *i.e.* heavy network architecture, inconsistent requirements about image

size. First of all, although it is an intuitive solution that adds an additional object detection branch to the existing two-branch gaze estimation models [4, 11, 21, 25, 30, 31, 36, 51], this approach would undoubtedly increase the number of calculations and parameters of the entire network. Besides, gaze estimation models usually employ an image of ordinary size (*e.g.* 224×224) to capture a global receptive field. In contrast, objects in the retail scenario are generally small and dense, requiring an enlarged image to detect bounding boxes precisely. Moreover, compared with individuals carrying out gaze estimation and object detection, a more suitable way should employ a unified framework and achieve joint optimization.

To alleviate the above issues, we make the following three designs. (1) Different from previous gaze estimation works [4, 21, 30] that use two independent branches (see in Fig. 1 (a)), we propose a specific-general-specific (SGS) mechanism to extract task-specific features from the scene and head image with only one backbone (see in Fig. 1 (b)), which can help to reduce the parameters and computational burdens, and also make it possible to joint optimization for different inputs. (2) To assist precise object detection, we develop a *Defocus* layer to generate object-specific features. In particular, the input image with ordinary size can not produce a feature map with sufficient resolution for detecting small and dense retail objects. The proposed *Defocus* layer zooms the spatial size via shrinking the channel size, which can produce a feature of high-resolution without losing information or bringing extra computations. (3) To tackle the performance bottleneck of imprecise gaze heatmap, we propose the energy aggregation loss that measures the percentage of energy within the stared box, and use the ground truth bounding box to guide the gaze estimation process.

In our work, we propose a unified framework, namely GaTector, to estimate gaze heatmap, detect retail objects and conduct gaze object detection, as shown in Figure 2. The scene and head images are first jointly tackled by the SGS feature extractor. Then, the object detection head discovers bounding boxes, and the gaze prediction head predicts gaze heatmap, so we can jointly consider gaze prediction and object detection results to carry out gaze object prediction. Also, a novel wUoC metrics proposed to better reveal the difference between boxes even when they share no overlapping area. Our contributions can be summarized as follows:

- We proposed a unified method GaTector with a novel wUoC evaluation metric to make an early exploration on the gaze object detection task.

- We propose a novel SGS mechanism that can extract task-specific features by a single backbone while maintaining satisfactory performance. A *Defocus* layer is introduced in SGS to prepare high-resolution feature maps for small retail object detection, and the

energy aggregation loss guides the gaze heatmap to be concentrated.

- On the large-scale GOO dataset, we consistently improve the performance on two traditional tracks, *i.e.* gaze estimation, and object detection, while reducing the model parameter and computational costs. In addition, we build a solid baseline for the gaze object detection task to promote future research.

2. Related Work

Gaze following. As a practical technique [45, 46], the gaze following task is proposed by Recasens *et al*. [30] and is a well-explored branch of gaze estimation. Gaze estimation reveals where a person is looking and serves as an essential clue for understanding human intention [41, 42]. Existing gaze estimation works can be divided into three categories according to different scenarios *i.e.*, gaze point estimation [24, 25], gaze following [3, 20, 39], and 3D gaze estimation [6, 26].

This paper is related to the gaze following task. In the early phase, Zhu *et al*. [56] proposed a unified model that can perform face detection, pose estimation, and landmark estimation in realistic images. Recasens *et al*. [30] predict the gaze area by extracting head pose and gaze direction via a deep model. Afterward, Parks *et al*. [28] combined saliency maps with human head pose and gaze direction to predict the gaze area for the observer. In addition, there are also some works focused on general gaze following [3, 27, 39]. For example, Mukherjee *et al*. [27] restored the interaction with the environment based on head pose estimation. Unlike predicting gaze area, this paper studies the gaze object prediction task and aims to discover the bounding box of the stared objects, which is more challenging.

Object detection. Recently, object segmentation [15, 16, 52, 53] and detection methods have received apparent progress, where detection methods can be divided into anchor-free detectors [2, 32–34, 37] and anchor-based detectors [8, 9, 22, 35, 50]. As a representative method for the anchor-free paradigm, YOLO [32] predicts bounding boxes at points close to the object's center. Later, a series of methods [2, 33, 34] based on YOLO gradually improved object detection performance and formed effective solutions to object detection. The anchor-based methods include two categories: one-stage methods [22, 23] and two-stage methods [13, 35]. The one-stage detector can directly make predictions based on the extracted feature map and default anchors, while the two-stage detector first generates object proposals and then performs detailed refinement.

Gaze prediction in the retail industry. In recent years, the automatic retail system and human-object interaction [54, 55] have aroused increasing research interests. For example, Harwood *et al*. [12] use mobile phones to track the visual attention of consumers in the retail environment. Be-
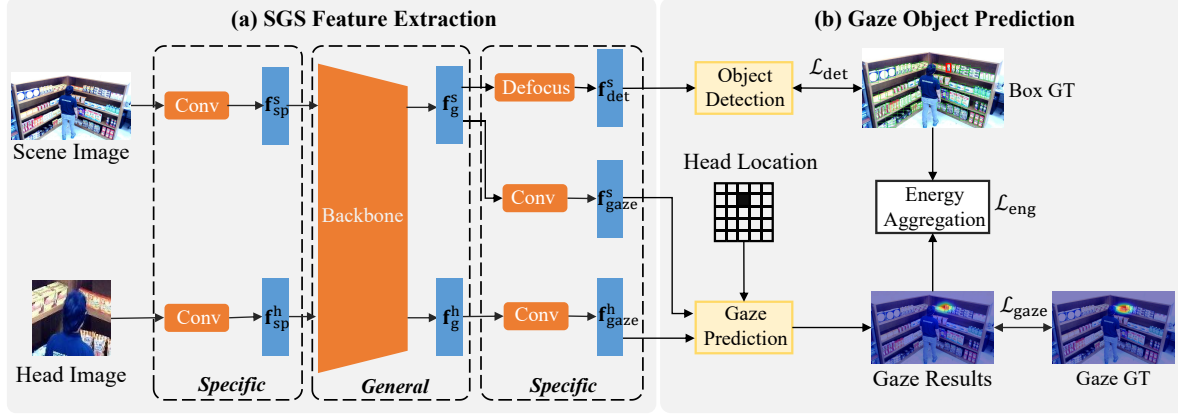
Figure 2. Overview of the proposed method. (a) The specific-general-specific (SGS) architecture can provide task-specific features while the sharing backbone. (b) The object detection module and gaze prediction module can be jointly trained by the energy aggregation loss.

sides, new benchmarks are proposed to serve the automatic checkout systems, *e.g.* the D2S dataset [10] and the RPC dataset [47]. Recently, EyeShopper [1] offered an innovative system to achieve a precise estimation of customers' sight. In the retail environment, it is valuable for estimating the stared products to perform exact recommendations. However, existing methods only predict gaze area and leave the gaze object prediction problem under-explored.

3. Method

Given a scene image $\mathbf{I}_s$ and head location mask $\mathbf{H}$, the head image $\mathbf{I}_h$ usually is generated by cropping the scene image $\mathbf{I}_s$. The goal of gaze object prediction task is to predict the bounding box and category label of the object started by a human.

3.1. SGS Mechanism

To extract holistic scene features and detailed head features, traditional gaze follow works [4, 21, 30] usually use two independent networks to tackle scene image $\mathbf{I}_s$ and head image $\mathbf{I}_h$, respectively. However, if this paradigm was used to resolve the gaze object detection problem, an extra object detection branch is needed and makes there exist three parallel branches in the model, which results in a significant increase of the parameters and computational cost. An intuitive way to reduce the parameters and computational cost is to share the backbone when extracting features for the object detection branch and gaze prediction branch. However, this manner will significantly reduce the performance (see Table 1) as the information required by the object detection and gaze prediction are different. To alleviate this issue, this paper proposes a specific-general-specific (SGS) mechanism to jointly consider the task specificity of scene image and head image before and after the shared backbone. SGS employs a sharing backbone to extract features in a general manner and utilizes specific heads to prepare task-specific features for gaze prediction and object detection.

As shown in Fig. 2 (a), we first use two independent convolutional layers to extract input-specific features for scene image and head image before the shared backbone:

$$\mathbf{f}_{sp}^s = \psi^s(\mathbf{I}_s), \ \ \mathbf{f}_{sp}^h = \psi^h(\mathbf{I}_h), \tag{1}$$

where $\psi^s(\cdot)$ and $\psi^h(\cdot)$ denote convolutional layers for scene image and head image, respectively. $\mathbf{f}_{sp}^s$ and $\mathbf{f}_{sp}^h$ denote the extracted features. Then we feed $\mathbf{f}_{sp}^s$ and $\mathbf{f}_{sp}^h$ into the shared backbone and produce features in a general manner:

$$\mathbf{f}_g^s = \psi^b(\mathbf{f}_{sp}^s), \ \ \mathbf{f}_g^h = \psi^b(\mathbf{f}_{sp}^h) \tag{2}$$

where $\mathbf{f}_g^s$ and $\mathbf{f}_g^s$ denote the general features for scene image and head image, ψ^b denotes the shared backbone. By sharing the backbone, the network parameters and computational cost would be significantly reduced. Afterward, we take the general scene feature $\mathbf{f}_g^s$ as input and design a novel *Defocus* layer (see in Section 3.3) to generate object-specific feature for detecting the location:

$$\mathbf{f}_{det}^s = \phi^{det}(\mathbf{f}_g^s) \tag{3}$$

where ϕ^{det} denotes the proposed *Defocus* layer, $\mathbf{f}_{det}^s$ denotes the feature for object detection. Then, both general head feature $\mathbf{f}_g^h$ and general scene feature $\mathbf{f}_g^s$ are sent into two independent convolution layers to generate gaze-specific features for predicting gaze results:

$$\mathbf{f}_{gaze}^s = \phi^s(\mathbf{f}_g^s), \ \ \mathbf{f}_{gaze}^h = \phi^h(\mathbf{f}_g^h), \tag{4}$$

where $\phi^s(\cdot)$ and $\phi^h(\cdot)$ indicate two independent convolution layers. $\mathbf{f}_{gaze}^s$ and $\mathbf{f}_{gaze}^h$ indicate the scene features and head features for the gaze prediction network (Section 3.2).

By sharing the backbone and generating specific features before and after the backbone, the proposed SGS mechanism can significantly reduce the parameters and computational cost while maintaining a satisfactory performance, forming an effective and efficient unified framework for gaze object detection.
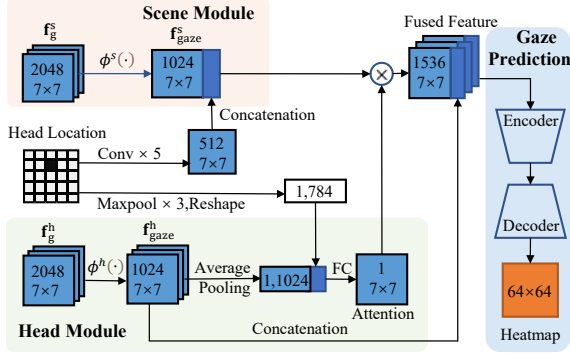
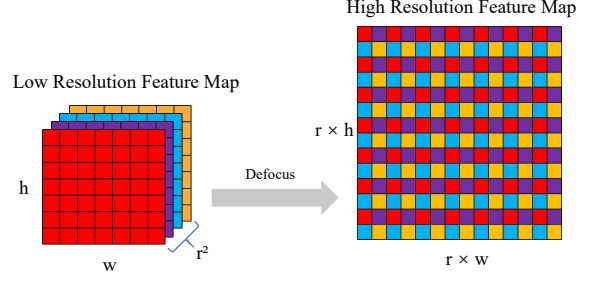Figure 3. Illustration of the gaze prediction network.

Figure 4. Illustration of the *Defocus* operation. One channel in the high-resolution feature map is transformed from r^2 channels in the low-resolution feature map.

3.2. Gaze prediction

As shown in Fig. 3, the gaze prediction network takes two gaze-specific features (*i.e.* $\mathbf{f}_{\text{gaze}}^{\text{s}}$ and $\mathbf{f}_{\text{gaze}}^{\text{h}}$) and the head location map (*i.e.* $\mathbf{H}$) as input to predict the gaze results.

As the head location map can provide valuable guidance for the gaze prediction task, Chong *et al.* [4] concatenate the head location map and scene image before extracting holistic scene features. However, the backbone in our GaTector serves both the gaze prediction task and the object detection task, which makes the head location may mislead the object detection process. Thus, we propose a novel "head-delay" strategy to resolve this problem. It first employs the backbone network to tackle the scene image $\mathbf{I}_{\text{s}}$ and produces the general feature. Then, the head location cues are supplied into the gaze prediction network. In particular, we use five convolutional layers with stride 2 to tackle the head location map and combine the generated feature with the scene feature.

Based on the above two improvements, we follow [4] to perform gaze object prediction. As shown in Fig. 3, the gaze prediction network mainly consists of three modules, *i.e.* the head module, the scene module, and the gaze prediction module. In the scene module, we first process the head location image $\mathbf{H}$ by five convolutional layers and concatenate the output with the gaze-specific scene feature $\mathbf{f}_{\text{gaze}}^{\text{s}}$ to make the feature map aware of the gaze position. Then, in the head module, we process the head location image $\mathbf{H}$ by three max-pooling with a flatten operation, then concatenate the output with the gaze-specific head feature $\mathbf{f}_{\text{gaze}}^{\text{s}}$ together to generate an attention map. Finally, we feed the element-wise product of this attention map and the output of the scene module into the gaze prediction module to get the gaze heatmap $\mathbf{M}$. Particularly, the gaze prediction module is an encoder-decoder structure, the encoder consists of two convolutional layers, and the decoder consists of three deconvolutional layers.

3.3. Object detection

As shown in Fig. 2, in the object detection branch, we propose a new layer named *Defocus* to generate object-specific feature $\mathbf{f}_{\text{det}}^{\text{s}}$. Then, we adopt the detection head of YOLOv4 [2] to discover objects in the retail scenario precisely.

Given an image, the backbone network tackles the entire scene image and outputs general features $\mathbf{f}_{\text{g}}^{\text{s}}$. An intuitive strategy to detect multiple small objects precisely in the retail scenario is to enlarge the scene image to generate high-resolution features. However, the enlarged scene images contribute little to the gaze prediction process and increase the computational burdens. Another strategy is to interpolate the feature map into a high resolution, which would lose some valuable information or bring extra computational costs. For example, considering a feature map $\mathbf{x} \in \mathbb{R}^{2048 \times 7 \times 7}$, it contains 2048-D features and exhibits spatial size 7×7. If we first compress the channel dimension via a convolutional layer and then interpolate the feature map to extend the dimension, some valuable information would be lost in the channel compressing process. Alternatively, if we directly interpolate the feature map by a factor of 2, the output would be $\mathbf{x}' \in \mathbb{R}^{2048 \times 14 \times 14}$, which increases computational costs for subsequent steps.

To resolve the problem above, we develop the *Defocus* layer to enlarge the feature map without losing information or taking extra computations. This *Defocus* layer is a reverse operation to the *focus* layer [5]. As shown in Fig. 4, given an enlarging ratio r, we first rearrange elements in the feature map and shrink the channel dimension by a factor of $1/r^2$, then zoom the height and width by a factor r. This rearranging operation requires fewer computational resources, and can hold all information in the produced object-specific feature $\mathbf{f}_{\text{det}}^{\text{s}}$.

After obtaining the object-specific feature $\mathbf{f}_{\text{det}}^{\text{s}}$, the detection network uses the feature pyramid structure to fuse features from different blocks, where feature maps are interpolated to pursue a consistent spatial size. We also replace the interpolation process with the proposed *Defocus* layer, which requires fewer computational sources but achieves precise detection results. Moreover, we find one high-resolution feature map is sufficient to detect retail objects and remove other detection heads, which saves 49.8% computational costs, as multiple objects in the retail scene
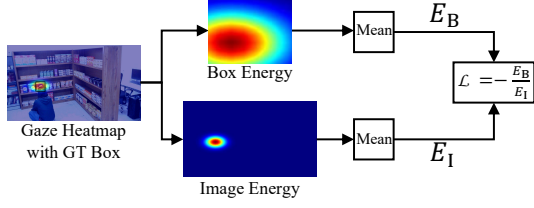
Figure 5. Calculation procedure of the energy aggregation loss. "Mean" denotes the element-wise mean operation.

do not exhibit drastic variations concerning the object size.

3.4. Energy aggregation loss

The studied gaze object prediction task usually requires high-quality gaze heatmaps to generate precise results. Take the model of [4] as an example, it achieves an AUC of 0.952 while the L_2 distance and angular error are 0.075 and 15.1°, respectively. Although numerical values of these two errors seem minor, such errors would cause apparent departure on the gaze point and lead to incorrect gaze object predictions. A high-quality gaze heatmap should accurately hit the gaze point, so its energy should be aggregated on the ground truth stared object. Thus, we propose the energy aggregation loss and utilize the ground truth to guide the gaze estimation process.

As shown in Fig. 5, given a heatmap $\mathbf{M}$ predicted by the gaze prediction module, the energy of each pixel can be represented as $\mathbf{M}_{i,j}$. If the ground truth gaze object box is defined as $\mathbf{b} = (x_1, y_1, x_2, y_2)$ where (x_1, y_1) and (x_2, y_2) are coordinates of the top left and bottom right corners, we can get the average energy within this box by:

$$E_b = \frac{1}{N} \sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} \mathbf{M}_{i,j}, \tag{5}$$

Where N indicates the number of elements within box $\mathbf{b}$. In addition, we can calculate the total energy within the image E_I by calculating the average value of all spatial bins. To guide the heatmap aggregated on the stared object box, we define the energy aggregation loss as:

$$\mathcal{L}_{eng} = -\frac{E_b}{E_I}. \tag{6}$$

3.5. Training and inference

Training. We assign three default boxes at each spatial location for object detection, whose spatial sizes are 12×16, 19×40, and 28×64, respectively. As our detection branch is based on the detection head of YOLOv4 [2], we employ the same method as YOLOv4 to calculate detection loss $\mathcal{L}_{det}$, which jointly considers the detection confidence score, category classification score, and bounding box regression. In the gaze prediction branch, we first generate the ground truth heatmap from the ground truth gaze point
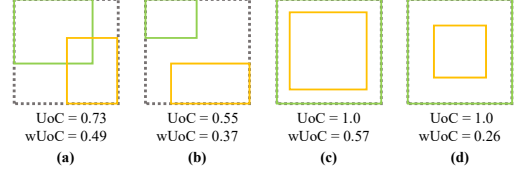

Figure 6. Illustration of our proposed metric. In cases (a) and (b), the wUoC metric gets small with the decrease of the overlapping area. In cases (c) and (d), wUoC can reveal the similarity when one box fully covers the other.

by applying Gaussian blur and then calculate the heatmap loss $\mathcal{L}_{gaze}$ via measuring the MSE loss between the predicted heatmap and the ground truth heatmap. In addition, the energy aggregation loss $\mathcal{L}_{eng}$ uses the ground truth object box to guide the training of the gaze prediction branch. The total loss used in the training process consists of these three terms:

$$\mathcal{L}_{total} = \mathcal{L}_{det} + \mathcal{L}_{gaze} + \mathcal{L}_{eng}. \tag{7}$$

Inference. In the inference process, the entire image and cropped head image are firstly processed by the proposed specific-general-specific mechanism to generate task-specific features. Then the gaze prediction branch estimates the gaze heatmap, and the object detection branch predicts the object bounding boxes. Afterward, we follow Eq. 5 to calculate the mean energy of each box and select the box with max mean energy as the predicted gaze object.

3.6. Evaluation metric

As a newly proposed task, our studied gaze object prediction requires a proper evaluation metric. The object detection community first measures the intersection over union (IoU) between predicted and ground-truth bounding boxes, then calculates the average precision (AP) to measure the performance. The AP score would be zero if there is no overlapping area (*i.e.* IoU=0), and AP cannot reveal the distance between prediction and ground truth. However, such distance is meaningful when performing gaze object prediction in the retail scenario. For example, the retail system first predicts the stared object and then sequentially recommends this object and its neighboring ones to the customer.

A reasonable solution is to make the traditional IoU metric aware of bounding box distances to alleviate this issue. Given the prediction box $\mathbf{p}$ and the ground truth box $\mathbf{g}$, we calculate their minimum closure and obtain a bounding box $\mathbf{a}$. As shown in Fig. 6 (a), the union over closure $\text{UoC} = \frac{\mathbf{p} \cup \mathbf{g}}{\mathbf{a}}$ can reveal the distance between prediction and ground truth. As shown in Fig. 6 (b), even if there is no overlapping, the UoC metric can also reveal the distance. However, when one box fully covers the other, the UoC metric would become 1, and cannot distinguish cases like Fig. 6 (c) and Fig. 6 (d).

Table 1. Gaze estimation performance on the GOO-Synth.

Setups	AUC ↑	Dist. ↓	Ang. ↓
Random	0.497	0.454	77.0
Recasens *et al.* [30]	0.929	0.162	33.0
Lian *et al.* [21]	0.954	0.107	19.7
Chong *et al.* [4]	0.952	0.075	15.1
#a Sharing backbone	0.905	0.139	27.1
#b Head-free scene branch	0.941	0.100	18.8
#c Head-delay	0.951	0.091	16.2
#d GaTector	**0.957**	**0.073**	**14.9**

Table 2. Object detection performance on the GOO-Real. The computational cost of each setup is reported in terms of Parameters (M) and FLOPs (G).

Setups	Para.	FLOPs	AP	AP$_{50}$	AP$_{75}$
#a YOLOv4 [2]	64.07	8.69	43.69	84.02	43.59
#b ResNet-50 backbone	61.65	7.85	40.87	82.04	37.48
#c Large feature map	40.35	**6.17**	41.07	81.67	38.46
#d Interpolation	40.35	12.37	54.67	94.93	58.21
#e Defocus	**39.53**	12.30	**56.20**	**96.90**	**60.23**
#f GaTector	60.78	18.11	52.25	91.92	55.34

We further introduce a size similarity weight into the UoC metric. The size similarity weight considers the area of two boxes and can be defined as $w = \min(\frac{\mathbf{p}}{\mathbf{g}}, \frac{\mathbf{g}}{\mathbf{p}})$. Thus, our proposed metric can be formulated as:

$$\text{wUoC} = \min\left(\frac{\mathbf{p}}{\mathbf{g}}, \frac{\mathbf{g}}{\mathbf{p}}\right) \times \frac{\mathbf{p} \cup \mathbf{g}}{\mathbf{a}}. \qquad (8)$$

As shown in Fig. 6 (c) and Fig. 6 (d), the wUoC metric can tackle the overlapping case and adaptively reveal the prediction quality.

4. Experiments

4.1. Setups

Dataset. We use the GOO dataset to evaluate our proposed method. GOO contains annotations of gaze point, gaze object, and bounding boxes for all objects, and has 24 different categories, where the GOO-Synth subset contains 192000 synthetic images. The GOO-Real subset contains 9552 real images. The GOO dataset exhibits the multiple challenges for the gaze object prediction task, such as small size objects, multiple objects. Images in the GOO dataset contain 80 objects on average.

Implementation details. In the proposed GaTector, the sharing backbone network adopts ResNet-50 [14]. As for object detection, we set $r = 2$ in the *Defocus* layer, use NMS with threshold 0.3 to remove redundant boxes, and keep the top 100 boxes for each image. As for gaze estimation, we apply Gaussian blur with kernel 3 to tackle the ground-truth gaze point. The network is optimized by the Adam [18] algorithm for 100 epochs. We set batch size as 32, and set the initial learning rate as 10^{-4}. All experiments are implemented based on the PyTorch [29] and GeForce RTX 3090 GPUs.

Metric. As for gaze object prediction, we use the wUoC metric as described in Section 3.6. For gaze prediction, we follow previous works [4, 21, 38] and use three metrics, *i.e.* AUC, L2 distance (Dist.), and Angular error (Ang.). The AUC metric determines whether each cell in the spatially discretized image is classified as a gaze object. Dist. is the L2 distance between the annotated gaze location and the prediction. Ang. is the angular error of the predicted

gaze. For object detection, we followed the previous works [2, 13, 35] to use the average precision (AP) as our metric.

4.2. Gaze prediction

In traditional gaze follow works [4, 21, 30], two separated networks are used to tackle the scene image and the head image, respectively. Under this paradigm, Lian *et al.* [21] and Chong *et al.* [4] achieve promising performance, as shown in Table 1. Aiming to employ a sharing backbone to perform gaze object detection, we study multiple strategies to integrate the scene and head branches.

First of all, we use ResNet-50 [14] as a sharing backbone and jointly employ the proposed SGS architecture to serve the scene and head branches. Following Chong *et al.* [4], we concatenate scene image and head location map and send the four-channel input to the scene branch. As shown in Table 1 setup #a, simply sharing backbone gets an AUC of 0.905 and leads to a 0.047 performance drop. The drop mainly comes from the random initialization of the convolutional block in the image branch. In particular, given a four-channel input, the first convolutional layer cannot use the pre-trained weights in ResNet-50 [14], and features extracted from the randomly initialized layer are insufficient to capture sensitive information.

Thus, we remove the head location map and only send the scene image to the scene branch. This experiment makes an AUC of 0.941 and verifies the necessity of proper initialization for the first convolutional layer. However, without the head location map, the scene branch still shows a performance drop concerning [4]. Consequently, we propose the head-delay strategy, as described in Section 3.2. The head-delay strategy gets comparable performance with Chong's method [4]. Moreover, the complete method of our proposed GaTector achieves high-quality gaze estimation performance and makes improvements under all three metrics.

4.3. Object detection

Table 2 studies the object detection performance under different setups. First of all, vanilla YOLOv4 [2] with CSP-Darknet53 initialization gets an AP of 43.69%, which requires 64.07M parameters. In the proposed GaTector, the

Figure 7. Visualization results of GaTector, we show the results of object detection (OD), gaze estimation (GE), gaze object prediction (GOD), and ground truth gaze object prediction (GOD GT). We provide five successful cases (left) and one fail case (right)

Table 3. Comparison of gaze object prediction performance, measured by wUoC (%) on the GOO-Real dataset.

Setups	Para.	FLOPs	AP_{50}	GOP GT gaze	GOP Pred. gaze
Faster-RCNN [35]	102.70	32.92	25.47	12.04	0.69
PAA [17]	93.39	19.40	54.72	13.61	1.01
RetinaNet [22]	98.05	20.02	72.86	15.06	1.77
Sabl [44]	97.64	19.58	73.10	15.14	1.21
FCOS [37]	93.34	19.16	74.78	14.17	0.66
YOLOv4 [2]	125.52	18.14	84.02	16.05	1.56
GaTector	**60.78**	**18.11**	**91.92**	**20.35**	**3.31**

object detection branch and the gaze prediction branch require the same backbone network. Thus, we use ResNet-50 [14] to replace the CSPDarknet53 backbone and, which leads to a performance drop of 2.82%. Considering the gaze object prediction dataset contains small objects with consistent size, we believe a large feature map is beneficial to detect gaze objects. Given three feature maps with different spatial sizes from YOLOv4, we only perform the object detection on the large feature map and observe slight performance improvements, as shown in Table 2 setup #c.

The small and dense objects inspire us to use large feature maps. However, due to the constraint of computational burdens and GPU memory, the spatial size of the input image is limited (*e.g.* 224 × 224). An intuitive way to enlarge the feature map is an interpolation, which gets an AP of 54.67% with the cost of more parameters and increased computations. In contrast to interpolation, the proposed *Defocus* layer can improve the performance to 56.2% but requires fewer computations. As for the complete method, GaTector makes an AP of 52.25% when only learns from the GOO-Real subset. The simplified data augmentation process primarily causes a performance gap be-

tween 56.20% and 52.25%. Specifically, when integrating gaze follow and object detection into a unified framework, we only keep transformations suitable to both tasks, *e.g.* random crop and color transformation. Because GOO-Real only contains limited images, the simplified data augmentation would damage the detection performance.

4.4. Gaze object prediction

Table 3 reports the gaze object detection performance under two different paradigms: separated paradigm and unified paradigm. Firstly, we can perform gaze object detection via two separated steps, *i.e.* object detection, and gaze selection. We carry out experiments with six well-performed methods: Faster-RCNN [35], PAA [17], RetinaNet [22], Sabl [44] FCOS [37] and YOLOv4 [2], including both anchor-based methods [2,17,22,35,44] and anchor-free methods [37], both one-stage methods [2, 17, 22, 37] and two-stage methods [35, 44]. Trained on the GOO-Real dataset, YOLOv4 [2] and FCOS [37] achieve precise performance while the FLOPs are limited.

Afterward, we employ a complete GOO dataset to train the proposed GaTector model and predict the gaze heatmap. Given object detection boxes and predicted gaze heatmaps, we follow Eq.(5) to select bounding boxes with high mean energy and report results for the gaze object prediction task. As shown in Table 3, although object detection results are acceptable, the gaze object prediction scores are low. We find the performance bottleneck lies in inaccurate gaze heatmaps. In contrast, if we utilize the ground truth heatmaps to select boxes from the detection results, the gaze object prediction performance would receive apparent gains.

Compared with the separated paradigm, our proposed GaTector adopts the unified paradigm, *i.e.* employing a uni-

Table 4. Ablation studies on GOO-Synth, we report the performance of gaze estimation (GE), object detection (OD), and gaze object prediction (GOP).

Setups	GE			OD			GOP
	AUC ↑	Dist. ↓	Ang. ↓	AP	AP_{50}	AP_{75}	wUoC
#a w/o input-specific blocks	0.596	0.387	70.17	35.2	66.4	41.3	0.0
#b w/o gaze-specific blocks	0.952	0.114	20.76	54.7	93.2	60.5	24.7
#c w/o object-specific Defocus	0.951	0.075	15.20	40.9	81.6	38.3	24.9
#d w/o $\mathcal{L}_{eng}$	0.946	0.103	17.20	55.8	94.4	61.5	25.2
GaTector	**0.957**	**0.073**	**14.91**	**56.8**	**95.3**	**62.5**	**28.5**

Table 5. Analysis about the bottleneck of GaTector on the GOO-Synth.

Gaze Heatmap		Object Box		GOP
GT	Pred.	GT	Pred.	mUoC
	✓		✓	28.50
	✓	✓		29.81
✓			✓	78.79

fied network to perform gaze follow, object detection, and conduct gaze object prediction. As shown in Table 3, GaTector consistently improves the object detection performance from 84.02% to 91.92%. However, due to the small amount of data in the Goo-Real dataset, current methods exhibit limited performance on the gaze object prediction task. There are two potential reasons, the predicted gaze point is not accurate enough, or our metric wUoC is stricter than traditional detection performance mAP.

4.5. Ablation studies

In GaTector, we propose an SGS feature extractor, a *Defocus* layer, and an energy aggregation loss to perform gaze object prediction. Table 4 reports ablation experiments about each component. Firstly, it would cause a dramatic performance drop by replacing two input-specific convolutional layers $\psi^s(\cdot)$ and $\psi^h(\cdot)$. This verifies the necessity of transposing inputs to a general space before sending them to the general backbone. Besides, both gaze follow and object detection receive an inevitable performance drop when removing gaze-specific output blocks $\phi^s(\cdot)$ and $\phi^h(\cdot)$, which verifies the influence of the task-specific output convolutional layer. In addition, we replace the *Defocus* layer with the traditional interpolation operation and observe performance drops. These three experiments prove the rationality and necessity of our proposed SGS feature extract mechanism. Finally, removing the energy aggregation loss would damage the performance of gaze estimation, especially under the metric L2 distance and angular error.

Table 5 analyzes the performance bottleneck of our proposed GaTector. Using predicted gaze heatmap and object bounding boxes, we select a suitable bounding box by Eq.(5) from prediction boxes within the heatmap, GaTec-

tor gets wUoC of 28.50%. If we use ground truth object boxes, the gaze object prediction performance can only be improved to 29.81%, which verifies the current approach has achieved accurate object detection performance. In contrast, the performance will receive huge improvements to 78.79%, if we use the ground truth heatmap to select boxes, which indicates that the current performance bottleneck mainly lies in gaze estimation. The bottleneck analysis provides a promising direction for future research.

4.6. Qualitative results

Fig. 7 exhibits qualitative results for the proposed GaTector method. Faced with various retail scenarios (*e.g.* various distances to the retail objects, different shooting views), our proposed GaTector can accurately localize bounding boxes and relatively precisely predict gaze heatmap, finally leading to accurate gaze object prediction results. The last row of Fig. 7 shows a failure case, where errors in the gaze heatmap mislead the gaze object prediction process.

5. Conclusion

We make an early exploration to build a unified framework to tackle the gaze object prediction task. To simultaneously conduct object detection and gaze estimation, we propose a novel SGS mechanism to extract two task-specific features with only one backbone, which can help to reduce model parameters and computational burdens. Then, we introduce two input-specific blocks before the shared backbone and three task-specific blocks after the shared backbone to consider the specificity of inputs. To serve small object detection in retail scenarios, we propose a *Defocus* layer to enlarge the feature map without losing information or bringing extra computations. Also, we design energy aggregation loss and employ the bounding box of the stared object to guide the gaze heatmap prediction. The promising gains of GaTector in all three tracks of the GOO dataset demonstrate the efficacy of our method, which would inspire related tasks, *e.g.* multi-modality learning [40], small object detection [9].

Limitations. Since our GaTector is only verified on the benchmark dataset, it may encounter challenges when applied to practical scenarios, *e.g.* lacking sufficient supervision [43, 49], online decision [48]. This indicates a promising research direction for future works.

References

[1] Carlos Bermejo, Dimitris Chatzopoulos, and Pan Hui. Eyeshopper: Estimating shoppers' gaze using cctv cameras. In *ACM MM*, pages 2765–2774, 2020. 3

[2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 2, 4, 5, 6, 7

[3] Rechele Brooks and Andrew N Meltzoff. The development of gaze following and its relation to language. *Developmental science*, 8(6):535–543, 2005. 2

[4] Eunji Chong, Yongxin Wang, Nataniel Ruiz, and James M Rehg. Detecting attended visual targets in video. In *CVPR*, pages 5396–5406, 2020. 1, 2, 3, 4, 5, 6

[5] YOLO Contributors. Yolov5. https://github.com/ultralytics/yolov5, 2021. 4

[6] Carlos Elmadjian, Pushkar Shukla, Antonio Diaz Tula, and Carlos H Morimoto. 3d gaze estimation in the scene volume with a head-mounted eye tracker. In *Proceedings of the Workshop on Communication by Gaze Interaction*, pages 1–9, 2018. 2

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

[8] Xiaoxu Feng, Junwei Han, Xiwen Yao, and Gong Cheng. Progressive contextual instance refinement for weakly supervised object detection in remote sensing images. *IEEE TGRS*, 58(11):8002–8012, 2020. 2

[9] Xiaoxu Feng, Junwei Han, Xiwen Yao, and Gong Cheng. Tcanet: Triple context-aware network for weakly supervised object detection in remote sensing images. *IEEE TGRS*, 2020. 2, 8

[10] Patrick Follmann, Tobias Bottger, Philipp Hartinger, Rebecca Konig, and Markus Ulrich. Mvtec d2s: densely segmented supermarket dataset. In *ECCV*, pages 569–585, 2018. 3

[11] Jian Guan, Liming Yin, Jianguo Sun, Shuhan Qi, Xuan Wang, and Qing Liao. Enhanced gaze following via object detection and human pose estimation. In *ICMM*, pages 502–513. Springer, 2020. 2

[12] Tracy Harwood and Martin Jones. Mobile eye-tracking in retail research. In *Current trends in eye tracking research*, pages 183–199. Springer, 2014. 2

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 2, 6

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6, 7

[15] Peiliang Huang, Junwei Han, Nian Liu, Jun Ren, and Dingwen Zhang. Scribble-supervised video object segmentation. *IEEE/CAA Journal of Automatica Sinica*, 9(2):339–353, 2021. 2

[16] Peiliang Huang, Junwei Han, Dingwen Zhang, and Mingliang Xu. Clrnet: Component-level refinement network for deep face parsing. *IEEE TNNLS*, 2021. 2

[17] Kang Kim and Hee Seok Lee. Probabilistic anchor assignment with iou prediction for object detection. In *ECCV*, 2020. 7

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 6

[19] Michael Land and Benjamin Tatler. *Looking and acting: vision and eye movements in natural behaviour*. Oxford University Press, 2009. 1

[20] Yunhao Li, Wei Shen, Zhongpai Gao, Yucheng Zhu, Guangtao Zhai, and Guodong Guo. Looking here or there? gaze following in 360-degree images. In *ICCV*, pages 3742–3751, 2021. 2

[21] Dongze Lian, Zehao Yu, and Shenghua Gao. Believe it or not, we know what you are looking at! In *ACCV*, pages 35–50. Springer, 2018. 1, 2, 3, 6

[22] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 2, 7

[23] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016. 2

[24] Yunfei Liu, Ruicong Liu, Haofei Wang, and Feng Lu. Generalizing gaze estimation with outlier-guided collaborative adaptation. In *ICCV*, pages 3835–3844, 2021. 2

[25] Yang Liu, Lei Zhou, Xiao Bai, Yifei Huang, Lin Gu, Jun Zhou, and Tatsuya Harada. Goal-oriented gaze estimation for zero-shot learning. In *CVPR*, pages 3794–3803, 2021. 2

[26] Benoit Massé, Stéphane Lathuilière, Pablo Mesejo, and Radu Horaud. Extended gaze following: Detecting objects in videos beyond the camera field of view. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pages 1–8. IEEE, 2019. 2

[27] Sankha S Mukherjee and Neil Martin Robertson. Deep head pose: Gaze-direction estimation in multimodal video. *IEEE TMM*, 17(11), 2015. 2

[28] Daniel Parks, Ali Borji, and Laurent Itti. Augmented saliency model using automatic 3d head pose detection and learned gaze following in natural scenes. *Vision research*, 116:113–126, 2015. 2

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *NeurIPS*, pages 8024–8035. Curran Associates, Inc., 2019. 6

[30] Adria Recasens, Aditya Khosla, Carl Vondrick, and Antonio Torralba. Where are they looking? *NeurIPS*, 28:199–207, 2015. 1, 2, 3, 6

[31] Adria Recasens, Carl Vondrick, Aditya Khosla, and Antonio Torralba. Following gaze in video. In *ICCV*, pages 1435–1443, 2017. 2

[32] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 2

[33] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, pages 7263–7271, 2017. 2

[34] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28:91–99, 2015. 2, 6, 7

[36] Akanksha Saran, Srinjoy Majumdar, Elaine Schaertl Short, Andrea Thomaz, and Scott Niekum. Human gaze following for human-robot interaction. In *IROS*, pages 8615–8621. IEEE, 2018. 2

[37] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, pages 9627–9636, 2019. 2, 7

[38] Henri Tomas, Marcus Reyes, Raimarc Dionido, Mark Ty, Jonric Mirando, Joel Casimiro, Rowel Atienza, and Richard Guinto. Goo: A dataset for gaze object prediction in retail environments. In *CVPR*, pages 3125–3133, 2021. 1, 6

[39] Jochen Triesch, Christof Teuscher, Gedeon O Deák, and Eric Carlson. Gaze following: why (not) learn it? *Developmental science*, 9(2):125–147, 2006. 2

[40] Binglu Wang, Le Yang, and Yongqiang Zhao. Polo: Learning explicit cross-modality fusion for temporal action localization. *IEEE Sign. Process. Letters*, 28:503–507, 2021. 8

[41] Binglu Wang, Xun Zhang, and Yongqiang Zhao. Exploring sub-action granularity for weakly supervised temporal action localization. *IEEE TCSVT*, 2021. 2

[42] Binglu Wang, Yongqiang Zhao, and Xuelong Li. Multiple instance graph learning for weakly supervised remote sensing object detection. *IEEE TGRS*, 60:1–12, 2021. 2

[43] Binglu Wang, Yongqiang Zhao, and Yani Zhang. Pfwnet: Pretraining neural network via feature jigsaw puzzle for weakly-supervised temporal action localization. *Neurocomputing*, 443:162–173, 2021. 8

[44] Jiaqi Wang, Wenwei Zhang, Yuhang Cao, Kai Chen, Jiangmiao Pang, Tao Gong, Jianping Shi, Chen Change Loy, and Dahua Lin. Side-aware boundary localization for more precise object detection. In *ECCV*, 2020. 7

[45] Shunzhou Wang, Tianfei Zhou, Yao Lu, and Huijun Di. Contextual transformation network for lightweight remote-sensing image super-resolution. *IEEE TGRS*, 60:1–13, 2022. 2

[46] Shunzhou Wang, Tianfei Zhou, Yao Lu, and Huijun Di. Detail-preserving transformer for light field image super-resolution. In *AAAI*, 2022. 2

[47] Xiu-Shen Wei, Quan Cui, Lei Yang, Peng Wang, and Lingqiao Liu. Rpc: A large-scale retail product checkout dataset. *arXiv preprint arXiv:1901.07249*, 2019. 3

[48] Le Yang, Junwei Han, and Dingwen Zhang. Colar: Effective and efficient online action detection by consulting exemplars. In *CVPR*, 2022. 8

[49] Le Yang, Junwei Han, Tao Zhao, Tianwei Lin, Dingwen Zhang, and Jianxin Chen. Background-click supervision for temporal action localization. *IEEE TPAMI*, 2021. 8

[50] Xiwen Yao, Xiaoxu Feng, Junwei Han, Gong Cheng, and Lei Guo. Automatic weakly supervised object detection from high spatial resolution remote sensing images via dynamic curriculum learning. *IEEE TGRS*, 59(1):675–685, 2020. 2

[51] Hao Zhao, Ming Lu, Anbang Yao, Yurong Chen, and Li Zhang. Learning to draw sight lines. *IJCV*, 128(5):1076–1100, 2020. 2

[52] Tianfei Zhou, Jianwu Li, Shunzhou Wang, Ran Tao, and Jianbing Shen. Matnet: Motion-attentive transition network for zero-shot video object segmentation. *IEEE TIP*, 29:8326–8338, 2020. 2

[53] Tianfei Zhou, Liulei Li, Xueyi Li, Chun-Mei Feng, Jianwu Li, and Ling Shao. Group-wise learning for weakly supervised semantic segmentation. *IEEE TIP*, 31:799–811, 2022. 2

[54] Tianfei Zhou, Siyuan Qi, Wenguan Wang, Jianbing Shen, and Song-Chun Zhu. Cascaded parsing of human-object interaction recognition. *IEEE TPAMI*, 2021. doi:10.1109/TPAMI.2021.3049156. 2

[55] Tianfei Zhou, Wenguan Wang, Siyuan Qi, Haibin Ling, and Jianbing Shen. Cascaded human-object interaction recognition. In *CVPR*, pages 4263–4272, 2020. 2

[56] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, pages 2879–2886. IEEE, 2012. 2