

# Supplementary Materials: AL-GTD: Deep Active Learning for Gaze Target Detection

The supplementary material expands on the implementation details of our proposed method, AL-GTD, and the other active learning (AL) methods used in comparisons (Sec. A). Additionally, we present the main paper results (Figs. 4 and 5) in tabular form (Sec. B) as well as additional ablation studies that were not a part of the main paper due to space limitations. Lastly, we present extra qualitative results (Sec. C).

## A IMPLEMENTATION DETAILS

**AL-GTD (Ours).** We resize the input to the scene ( $S$ ), depth ( $D$ ), and head ( $H$ ) branches to  $224 \times 224$ , while the output gaze heatmap has a size of  $64 \times 64$ . We perform up to 15 training epochs for each AL iteration and evaluate the model performance for every 5 epochs. In all our experiments, we independently train multiple times using the same initial split and apply identical augmentations to each method.

**Other AL methods.** Except for VAAL [11] and LL4L [15], whose implementation requires additional specific losses, all other methods employ the same loss formulation. The implementation details of the other AL methods adopted for the gaze target detection task are as follows.

**(1) Entropy Sampling [10].** We apply the softmax operator to the predicted gaze heatmap  $H_G$  with a temperature  $\tau = 0.05$  and calculate the resulting distribution’s entropy. The acquisition function prioritizes unlabeled samples based on uncertainty, labeling the top- $\beta/N$  samples.

**(2) MC-Dropout [6].** Following the insights of Gal *et al.* [6], i.e. neural networks with dropout are equivalent to a probabilistic deep Gaussian process, we incorporate dropout layers into *GTN* after  $S$ ,  $D$ ,  $H$ , and within  $D_H$ . The sample score is the average pixel-wise variance of the gaze heatmaps generated through 16 samplings with dropout enabled.

**(3) LL4AL [15].** We adopt the formulation of the *loss prediction module* of Yoo *et al.* [15], and we extend our *GTN* with a learnable module that predicts the target loss of unlabeled samples. The loss module is trained jointly with *GTN*, using a scale-invariant loss function. As such, the update loss function of *GTN* with LL4L is:

$$L_{total}(\mathbb{A}) = \sum_{a \in \mathbb{A}} L_h(a) + \sum_{a \in \mathbb{A}} L_{LL4AL}(a) \quad (9)$$

with  $L_{LL4AL}$  being the loss function defined in [15]. The sample score is the output of the *loss prediction module*.

**(4) VAAL [11].** Sinha *et al.* [11] integrated a Variational AutoEncoder (VAE) and a discriminator to learn an informativeness score for unlabeled samples. We adapt the original implementation, where the VAE reconstructs the original input image, to suit the gaze target detection task, ensuring the representativeness of an image

**Table 6: AUC of our method AL-GTD with SOTA AL methods on the GazeFollow [9] dataset.**

Method	3.7K	6.2K	8.7K	11.2K	13.7K
Random	82.64 ± 0.00	83.49 ± 0.18	85.18 ± 0.40	86.12 ± 0.54	86.47 ± 0.30
Entropy [10]	82.64 ± 0.00	83.70 ± 0.60	84.44 ± 0.20	84.52 ± 1.27	85.80 ± 1.65
MC-Dropout [6]	82.64 ± 0.00	82.99 ± 0.18	83.97 ± 0.09	84.64 ± 0.77	86.29 ± 0.69
LL4AL [15]	80.51 ± 0.00	82.67 ± 0.33	83.81 ± 0.21	84.89 ± 0.59	85.00 ± 0.56
VAAL [11]	81.12 ± 0.00	83.28 ± 0.31	84.64 ± 0.43	85.61 ± 0.33	85.79 ± 0.20
UnReGa [2]	82.17 ± 0.00	84.35 ± 0.29	84.65 ± 0.39	85.70 ± 0.29	86.33 ± 0.47
AL-SSL [4]	82.64 ± 0.00	83.46 ± 0.49	85.32 ± 0.15	85.71 ± 0.24	86.62 ± 0.12
<b>AL-GTD (Ours)</b>	<b>82.64 ± 0.00</b>	<b>85.10 ± 0.74</b>	<b>85.81 ± 0.23</b>	<b>87.14 ± 0.21</b>	<b>87.67 ± 0.31</b>

**Table 7: Average distance of our method AL-GTD with SOTA AL methods on the GazeFollow [9] dataset.**

Method	3.7K	6.2K	8.7K	11.2K	13.7K
Random	0.260 ± 0.000	0.247 ± 0.001	0.234 ± 0.002	0.223 ± 0.005	0.217 ± 0.005
Entropy [10]	0.260 ± 0.000	0.252 ± 0.002	0.238 ± 0.004	0.245 ± 0.018	0.229 ± 0.011
MC-Dropout [6]	0.260 ± 0.000	0.254 ± 0.003	0.244 ± 0.002	0.240 ± 0.017	0.221 ± 0.005
LL4AL [15]	0.272 ± 0.000	0.259 ± 0.002	0.246 ± 0.002	0.238 ± 0.005	0.229 ± 0.009
VAAL [11]	0.274 ± 0.000	0.254 ± 0.005	0.234 ± 0.006	0.226 ± 0.001	0.216 ± 0.004
UnReGa [2]	0.261 ± 0.000	0.249 ± 0.004	0.234 ± 0.004	0.225 ± 0.005	0.218 ± 0.004
AL-SSL [4]	0.260 ± 0.000	0.244 ± 0.005	0.225 ± 0.003	0.220 ± 0.002	0.212 ± 0.003
<b>AL-GTD (Ours)</b>	<b>0.260 ± 0.000</b>	<b>0.237 ± 0.010</b>	<b>0.218 ± 0.006</b>	<b>0.214 ± 0.004</b>	<b>0.208 ± 0.002</b>

using both scene and head inputs. As such, we aim to reconstruct the gaze heatmap  $H$  with a Variational Autoencoder (VAE) while training a discrimination function between labeled and unlabeled samples. The final loss function is defined as

$$L_{total}(\mathbb{A}) = \sum_{a \in \mathbb{A}} L_h(a) + \sum_{a \in \mathbb{A}} L_{VAAL}(a) \quad (10)$$

with  $L_{VAAL}$  being the loss function defined in [11].

**(5) AL-SSL [4].** We adopt AL-SSL to the gaze target detection task by estimating the inconsistency  $inc = ||H_G - \mathcal{A}^{-1}(H'_G)||_2$  between the original and flipped gaze heatmap, where highest inconsistency corresponds to samples to be manually labeled by the oracle. Pseudo-labeling follows the approach outlined in our method (Sec. 3.2.2), with a pseudo-labeling percentage of 2%.

**(6) UnReGa [2].** We leverage the uncertainty definition from UnReGa to establish a committee-based AL method. This involves bootstrapping ensemble models  $GTN_i^t$  from the trained set at various epochs  $t$  from the current AL iteration  $i$ . The acquisition function is based on the uncertainty of the ensemble models, where a higher value indicates increased uncertainty and a more informative sample.

## B RESULTS

Tables 6, 7, 8, 9 and 10 showcase the results of our proposed method, AL-GTD, alongside SOTA AL methods on GazeFollow [9] and VideoAttentionTarget [3]. These results correspond to the figures presented in the main paper, specifically Figs. 4 and 5.

**Table 8: Minimum distance of our method AL-GTD with SOTA AL methods on the GazeFollow [9] dataset.**

Method	3.7K	6.2K	8.7K	11.2K	13.7K
Random	<b>0.185 ± 0.000</b>	0.174 ± 0.002	0.162 ± 0.002	0.152 ± 0.005	0.146 ± 0.005
Entropy [10]	<b>0.185 ± 0.000</b>	0.179 ± 0.003	0.166 ± 0.003	0.172 ± 0.017	0.158 ± 0.011
MC-Dropout [6]	<b>0.185 ± 0.000</b>	0.180 ± 0.003	0.171 ± 0.002	0.167 ± 0.016	0.148 ± 0.005
LL4AL [15]	0.199 ± 0.000	0.186 ± 0.002	0.173 ± 0.003	0.165 ± 0.004	0.158 ± 0.009
VAAL [11]	0.200 ± 0.000	0.180 ± 0.005	0.162 ± 0.006	0.154 ± 0.001	0.145 ± 0.003
UnReGa [2]	0.187 ± 0.000	0.176 ± 0.004	0.162 ± 0.005	0.153 ± 0.004	0.147 ± 0.004
AL-SSL [4]	<b>0.185 ± 0.000</b>	0.171 ± 0.005	0.153 ± 0.003	0.149 ± 0.002	0.142 ± 0.002
<b>AL-GTD (Ours)</b>	<b>0.185 ± 0.000</b>	<b>0.166 ± 0.010</b>	<b>0.148 ± 0.005</b>	<b>0.144 ± 0.004</b>	<b>0.140 ± 0.006</b>

**Table 9: AUC of our method AL-GTD with SOTA AL methods on the VideoAttentionTarget [3] dataset.**

Method	932	1.9K	2.8K	3.7K	4.7K
Random	88.73 ± 0.00	88.80 ± 0.14	88.98 ± 0.26	89.15 ± 0.29	89.41 ± 0.34
Entropy [10]	88.73 ± 0.00	88.99 ± 0.13	89.08 ± 0.16	89.26 ± 0.36	89.56 ± 0.48
MC-Dropout [6]	88.73 ± 0.00	88.70 ± 0.16	88.94 ± 0.34	89.13 ± 0.24	89.37 ± 0.27
LL4AL [15]	<b>88.77 ± 0.00</b>	88.90 ± 0.07	89.07 ± 0.19	89.23 ± 0.46	89.39 ± 0.39
VAAL [11]	88.40 ± 0.00	88.02 ± 0.02	87.93 ± 0.15	88.17 ± 0.28	88.42 ± 0.35
UnReGa [2]	88.43 ± 0.00	88.60 ± 0.02	88.77 ± 0.02	89.06 ± 0.19	89.32 ± 0.27
AL-SSL [4]	88.62 ± 0.00	88.63 ± 0.14	88.95 ± 0.17	89.35 ± 0.13	89.54 ± 0.21
<b>AL-GTD (Ours)</b>	<b>88.73 ± 0.00</b>	<b>89.01 ± 0.06</b>	<b>89.48 ± 0.12</b>	<b>89.70 ± 0.14</b>	<b>90.03 ± 0.24</b>

**Table 10: Average distance of our method AL-GTD with SOTA AL methods on the VideoAttentionTarget [3] dataset.**

Method	932	1.9K	2.8K	3.7K	4.7K
Random	0.199 ± 0.000	0.196 ± 0.001	0.191 ± 0.002	0.190 ± 0.002	0.188 ± 0.003
Entropy [10]	0.199 ± 0.000	0.196 ± 0.000	0.191 ± 0.002	0.187 ± 0.004	0.183 ± 0.004
MC-Dropout [6]	0.199 ± 0.000	0.197 ± 0.002	0.196 ± 0.002	0.191 ± 0.003	0.189 ± 0.002
LL4AL [15]	0.186 ± 0.000	0.186 ± 0.001	0.186 ± 0.002	0.185 ± 0.003	0.184 ± 0.002
VAAL [11]	0.192 ± 0.000	0.196 ± 0.000	0.197 ± 0.003	0.197 ± 0.003	0.194 ± 0.004
UnReGa [2]	0.195 ± 0.000	0.191 ± 0.001	0.189 ± 0.001	0.185 ± 0.001	0.184 ± 0.002
AL-SSL [4]	0.187 ± 0.000	0.188 ± 0.000	0.188 ± 0.001	0.184 ± 0.001	0.182 ± 0.001
<b>AL-GTD (Ours)</b>	<b>0.181 ± 0.000</b>	<b>0.181 ± 0.001</b>	<b>0.181 ± 0.002</b>	<b>0.180 ± 0.002</b>	<b>0.177 ± 0.002</b>

In terms of AUC, averaging performance across AL cycles, the method ranking for GazeFollow is (Tab. 6): AL-GTD (ours), AL-SSL [4], random, UnReGa [2], MC-Dropout [6], VAAL [11], Entropy [10], and LL4AL [15]. For VideoAttentionTarget, the order is (Tab. 9): AL-GTD (ours), Entropy [10], AL-SSL [4], random, LL4AL [15], MC-Dropout [6], UnReGa [2], and VAAL [11].

Regarding average distance, the ranking for GazeFollow is (Tab. 7): AL-GTD (ours), AL-SSL [4], VAAL [11], random, UnReGa [2], MC-Dropout [6], Entropy [10], and LL4AL [15]. For VideoAttentionTarget, it is (Tab. 10): AL-GTD (ours), AL-SSL [4], Entropy [10], UnReGa [2], LL4AL [15], random, MC-Dropout [6], and VAAL [11]. Lastly, regarding minimum distance, the ranking for GazeFollow is (Tab. 7): AL-GTD (ours), AL-SSL [4], VAAL [11], random, UnReGa [2], MC-Dropout [6], Entropy [10], and LL4AL [15].

It is crucial to note that when there is noticeable diversity among the images in a dataset, it allows for a better assessment and comparison of the performance of AL methods. In the context of video datasets like VideoTargetAttention, where many frames and gaze annotations are similar, distinguishing between AL methods becomes more challenging, and random selection shows marginally better performance w.r.t. image datasets like GazeFollow.

**Ablation study.** Following Tab. 1 of the main paper, Table 11 and 12 report the effectiveness of SSL and the components of the AL acquisition function for the average and minimum distance

of the GazeFollow dataset. We first understand the effect of SSL by removing it from our pipeline (Row 1 vs. Row 6), which shows an increase in average and minimum distances on all AL cycles. Furthermore, (Rows 2-4) we show the impact of each acquisition component. The results show that the full model performs best while, on average, removing the objectness ( $\Gamma$ ) criterion results in a slightly greater increase compared to removing the others.

**Effect of SSL.** Tabs. 13 and 14 extend the results of Tab. 4 of the main paper, presenting the average and minimum distances for the GazeFollow dataset at each AL cycle, respectively. Consistent with the findings in the main paper (Sec. 4.2.4), AL-GTD outperforms all the SSL-injected counterparts, and SSL-based [4].

**Impact of pseudo-labeling.** Tab. 15 shows the effect of pseudo-labeling on the VideoAttentionTarget dataset, confirming the findings of the main paper (Sec. 4.2.3). The optimal performance is achieved when the amount of samples pseudo-labeled is equal to the number of samples labeled by the oracle.

**Comparison with SOTA Gaze Target Detectors.** Tab. 16 compares the performance of AL-GTD with SOTA gaze target detectors [12–14] in a low-data regime scenario, utilizing the dataset selected by AL-GTD (Row 2-4). It is essential to emphasize that for the VideoAttentionTarget dataset, the standard approach adopted by all studies utilizing it such as [1, 3, 5, 7, 8, 12–14] is to fine-tune their model with VideoAttentionTarget dataset after pre-training it on the GazeFollow dataset. In other words, all gaze target detection results are achieved by initially training the models on GazeFollow and then on the VideoAttentionTarget dataset. Taking this into account, our study also adheres to this methodology. Consequently, it becomes more challenging to compare SOTA gaze target detectors under a low-data regime since they have access to more data compared to the experiments performed with the GazeFollow dataset. This surely brings in an advantage to the transformer-based SOTA [13, 14] as well as allowing the [12] to close the performance gap with the proposed method. Nevertheless, the results once again confirm the better performance of AL-GTD in terms of all metrics.

**Inference time of AL-GTD and other AL methods.** Tab. 17 shows the number of parameters and the elapsed times for training, AL, and inference, for AL-GTD and other AL methods. Our AL-GTD requires fewer parameters (only 92.2M) compared to LL4AL [15], VAAL [11], and UnReGa [2], which rely on additional modules. Our running time is comparable to other AL methods like AL-SSL [4].

## C QUALITATIVE RESULTS

Figs. 6 and 7 showcase supplementary qualitative results comparing the performance of AL-GTD against random, Entropy [10], VAAL [11], LL4AL [15], UnReGa [2], and AL-SSL [4], showing that our method can generate superior gaze heatmaps, whereas other methods tend to produce heatmaps with multiple modes or sparser activation points.

**Table 11: Ablation study on GazeFollow [9] in terms of average distance (best in bold, lower is better) for the effect of SSL, and the components of the AL acquisition function: objectness ( $\Gamma$ ), discrepancy ( $\Delta$ ), and scatteredness ( $\Sigma$ ). We also report the results of random sampling and AL-SSL [4] for the reader’s reference. Note that the cycle with 3.7K samples represents the initial training, where no AL is applied. Therefore, the results are equal for all.**

SSL	$\Gamma$	$\Delta$	$\Sigma$	3.7K	6.2K	8.7K	11.2K	13.7K
✗	✓	✓	✓	<b>0.260 ± 0.000</b>	0.252 ± 0.004	0.238 ± 0.006	0.235 ± 0.008	0.228 ± 0.009
✓	✗	✓	✓	<b>0.260 ± 0.000</b>	0.242 ± 0.003	0.228 ± 0.006	0.218 ± 0.004	0.213 ± 0.004
✓	✓	✗	✓	<b>0.260 ± 0.000</b>	0.242 ± 0.007	0.223 ± 0.002	0.215 ± 0.003	0.206 ± 0.002
✓	✓	✓	✗	<b>0.260 ± 0.000</b>	0.242 ± 0.005	0.227 ± 0.005	0.214 ± 0.005	0.209 ± 0.004
✓	✓	✗	✗	<b>0.260 ± 0.000</b>	0.240 ± 0.005	0.224 ± 0.003	0.214 ± 0.008	0.220 ± 0.019
✓	✓	✓	✓	<b>0.260 ± 0.000</b>	<b>0.237 ± 0.010</b>	<b>0.218 ± 0.006</b>	<b>0.214 ± 0.004</b>	<b>0.208 ± 0.002</b>
Random				<b>0.260 ± 0.000</b>	0.247 ± 0.001	0.234 ± 0.002	0.223 ± 0.005	0.217 ± 0.005
AL-SSL [4]				<b>0.260 ± 0.000</b>	0.244 ± 0.005	0.225 ± 0.003	0.220 ± 0.002	0.212 ± 0.003

**Table 12: Ablation study on GazeFollow [9] in terms of minimum distance (best in bold, lower is better) for the effect of SSL, and the components of the AL acquisition function: objectness ( $\Gamma$ ), discrepancy ( $\Delta$ ), and scatteredness ( $\Sigma$ ). We also report the results of random sampling and AL-SSL [4] for the reader’s reference. Note that the cycle with 3.7K samples represents the initial training, where no AL is applied. Therefore, the results are equal for all.**

SSL	$\Gamma$	$\Delta$	$\Sigma$	3.7K	6.2K	8.7K	11.2K	13.7K
✗	✓	✓	✓	<b>0.185 ± 0.000</b>	0.179 ± 0.003	0.167 ± 0.006	0.163 ± 0.007	0.157 ± 0.008
✓	✗	✓	✓	<b>0.185 ± 0.000</b>	0.169 ± 0.003	0.157 ± 0.006	0.148 ± 0.003	0.143 ± 0.002
✓	✓	✗	✓	<b>0.185 ± 0.000</b>	0.170 ± 0.006	0.152 ± 0.003	0.145 ± 0.002	0.137 ± 0.001
✓	✓	✓	✗	<b>0.185 ± 0.000</b>	0.170 ± 0.004	0.157 ± 0.005	0.145 ± 0.004	0.139 ± 0.004
✓	✓	✗	✗	<b>0.185 ± 0.000</b>	0.168 ± 0.004	0.155 ± 0.002	0.145 ± 0.007	0.150 ± 0.017
✓	✓	✓	✓	<b>0.185 ± 0.000</b>	<b>0.166 ± 0.010</b>	<b>0.148 ± 0.005</b>	<b>0.144 ± 0.004</b>	<b>0.140 ± 0.006</b>
Random				<b>0.185 ± 0.000</b>	0.174 ± 0.002	0.162 ± 0.002	0.152 ± 0.005	0.146 ± 0.005
AL-SSL [4]				<b>0.185 ± 0.000</b>	0.171 ± 0.005	0.153 ± 0.003	0.149 ± 0.002	0.142 ± 0.002

**Table 13: Comparisons of average distance on GazeFollow among SSL-based methods for different cycles of AL. Note that the cycle with 3.7K samples represents the initial training stage, where no AL is applied. Thus, the results are equal for methods whose loss function is the same.**

Method + SSL	3.7K	6.2K	8.7K	11.2K	13.7K
Entropy [10]	<b>0.260 ± 0.000</b>	0.242 ± 0.009	0.231 ± 0.003	0.238 ± 0.008	0.218 ± 0.011
MC-Dropout [6]	<b>0.260 ± 0.000</b>	0.241 ± 0.003	0.225 ± 0.004	0.217 ± 0.004	0.226 ± 0.024
UnReGa [2]	0.259 ± 0.000	0.242 ± 0.005	0.221 ± 0.006	0.217 ± 0.002	0.211 ± 0.004
AL-SSL [4]	<b>0.260 ± 0.000</b>	0.244 ± 0.005	0.225 ± 0.003	0.220 ± 0.002	0.212 ± 0.003
AL-GTD (Ours)	<b>0.260 ± 0.000</b>	<b>0.237 ± 0.010</b>	<b>0.218 ± 0.006</b>	<b>0.214 ± 0.004</b>	<b>0.208 ± 0.002</b>

**Table 14: Comparisons of minimum distance on GazeFollow among SSL-based methods for different cycles of AL. Note that the cycle with 3.7K samples represents the initial training stage, where no AL is applied. Thus, the results are equal for methods whose loss function is the same.**

Method + SSL	3.7K	6.2K	8.7K	11.2K	13.7K
Entropy [10]	<b>0.185 ± 0.000</b>	0.170 ± 0.008	0.159 ± 0.003	0.165 ± 0.007	0.148 ± 0.010
MC-Dropout [6]	<b>0.185 ± 0.000</b>	0.169 ± 0.003	0.154 ± 0.005	0.145 ± 0.003	0.154 ± 0.023
UnReGa [2]	0.187 ± 0.000	0.171 ± 0.004	0.152 ± 0.005	0.144 ± 0.002	0.141 ± 0.002
AL-SSL [4]	<b>0.185 ± 0.000</b>	0.171 ± 0.005	0.153 ± 0.003	0.149 ± 0.002	0.142 ± 0.002
AL-GTD (Ours)	<b>0.185 ± 0.000</b>	<b>0.166 ± 0.010</b>	<b>0.148 ± 0.005</b>	<b>0.144 ± 0.004</b>	<b>0.140 ± 0.006</b>

**Table 15: Performance of AL-GTD associated with different numbers of samples pseudo-labeled. The results correspond to 4.7K manually annotated samples from the VideoAttentionTarget [3] dataset.**

Percentage	AUC ↑	Avg. Dist. ↓
0%	89.41 ± 0.15	0.180 ± 0.002
1%	89.82 ± 0.23	0.179 ± 0.004
2%	<b>90.03 ± 0.24</b>	<b>0.177 ± 0.002</b>

**Table 16: Comparisons between our AL-GTD and SOTA gaze target detectors trained under low data regimes (i.e. ~3% and ~40% of the dataset, respectively) on VideoAttentionTarget [3] using our AL-GTD’s sample selection.**

	AUC ↑		Avg. Dist. ↓	
	3%	40%	3%	40%
[12] (ICMI 2022)	88.5	92.1	0.189	0.128
[14] (CVPR 2022)	82.7	89.8	0.204	0.128
[13] (ICCV 2023)	88.2	91.3	0.203	0.123
AL-GTD (Ours)	<b>89.0</b>	<b>93.5</b>	<b>0.181</b>	<b>0.119</b>

**Table 17: Comparisons among AL-GTD and other AL methods in terms of number of parameters, and training, AL, and inference times per iteration.**

Method	# of params	Training (sec/iter)	AL (sec/iter)	Inference (sec/iter)
Random	92.2M	0.004	0.006	0.004
Entropy [10]	92.2M	<b>0.006</b>	<b>0.007</b>	0.005
MC-Dropout [6]	92.2M	0.008	0.021	0.013
LL4AL [15]	92.7M	0.009	0.012	0.006
VAAL [11]	106.3M	0.008	0.016	0.008
UnReGa [2]	276.6M	0.118	0.035	0.030
AL-SSL [4]	92.2M	0.017	0.008	<b>0.003</b>
<b>AL-GTD (Ours)</b>	<b>92.2M</b>	0.017	0.017	0.011





Figure 6: Gaze heatmaps produced by AL-GTD and others.



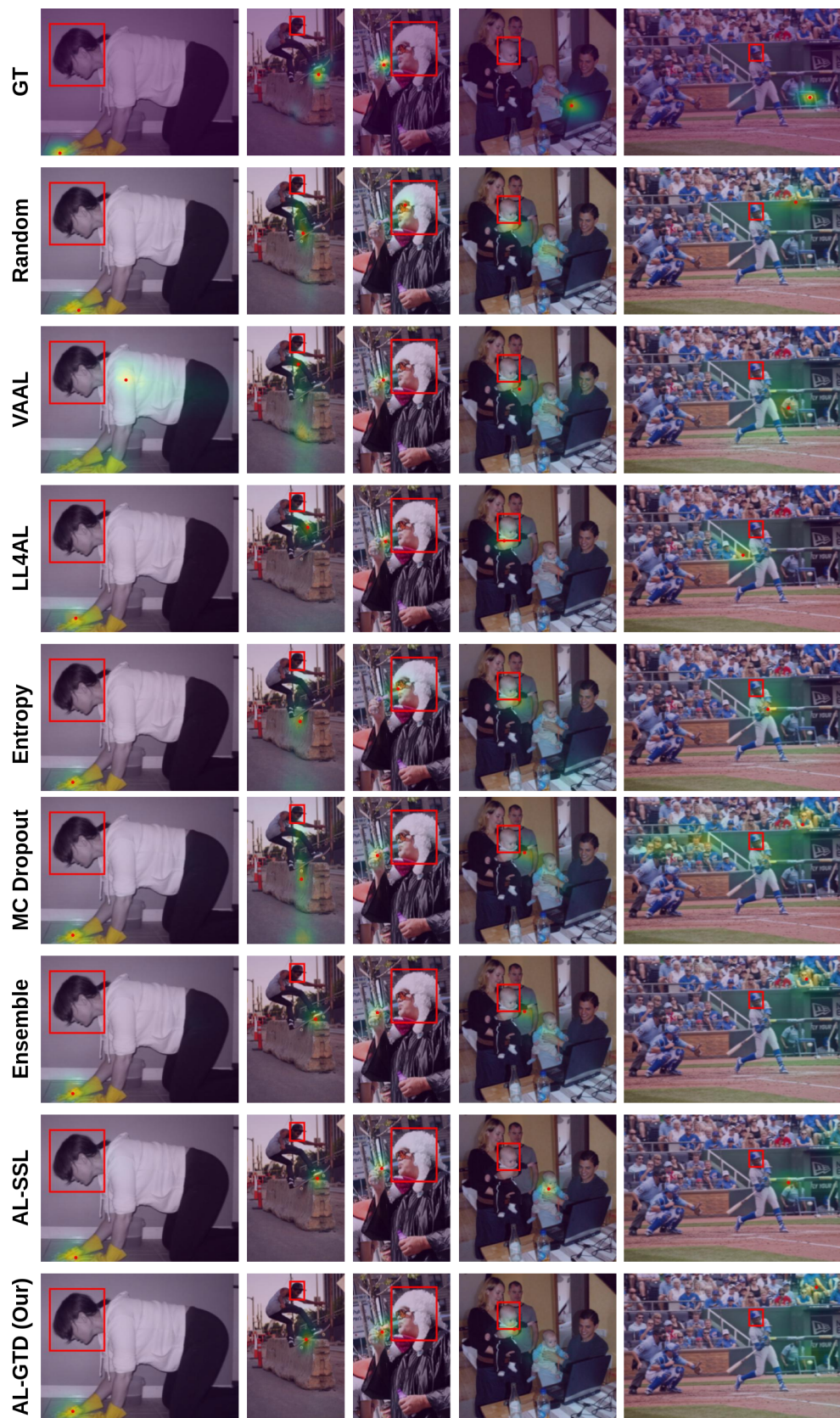


Figure 7: Gaze heatmaps produced by AL-GTD and others.

## REFERENCES

- [1] Jun Bao, Buyu Liu, and Jun Yu. 2022. ESCNet: Gaze Target Detection with the Understanding of 3D Scenes. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, New Orleans, LA, USA, 14126–14135.
- [2] Xin Cai, Jiabei Zeng, Shiguang Shan, and Xilin Chen. 2023. Source-Free Adaptive Gaze Estimation by Uncertainty Reduction. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Vancouver, BC, Canada, 22035–22045.
- [3] Eunji Chong, Yongxin Wang, Nataniel Ruiz, and James M Rehg. 2020. Detecting attended visual targets in video. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Computer Vision Foundation / IEEE, Seattle, WA, USA, 5396–5406.
- [4] Ismail Elezi, Zhiding Yu, Anima Anandkumar, Laura Leal-Taixe, and Jose M Alvarez. 2022. Not all labels are equal: Rationalizing the labeling costs for training object detection. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, New Orleans, LA, USA, 14492–14501.
- [5] Yi Fang, Jiapeng Tang, Wang Shen, Wei Shen, Xiao Gu, Li Song, and Guangtao Zhai. 2021. Dual attention guided gaze target detection in the wild. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Computer Vision Foundation / IEEE, Virtual, 11390–11399.
- [6] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *International conference on machine learning*. PMLR, PMLR, Sydney, NSW, Australia, 1183–1192.
- [7] Tianlei Jin, Qizhi Yu, Shiqiang Zhu, Zheyuan Lin, Jie Ren, Yuanhai Zhou, and Wei Song. 2022. Depth-aware gaze-following via auxiliary networks for robotics. *Engineering Applications of Artificial Intelligence* 113 (2022), 104924.
- [8] Qiaomu Miao, Minh Hoai, and Dimitris Samaras. 2023. Patch-level Gaze Distribution Prediction for Gaze Following. In *Proc. of the IEEE/CVF Winter Conference on Applications of Computer Vision*. IEEE, Waikoloa, HI, USA, 880–889.
- [9] Adria Recasens, Aditya Khosla, Carl Vondrick, and Antonio Torralba. 2015. Where are they looking?. In *Advances in Neural Information Processing Systems*, Vol. 28. Curran Associates, Inc., Montreal, Quebec, Canada, 199–207.
- [10] Claude Elwood Shannon. 2001. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review* 5, 1 (2001), 3–55.
- [11] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. 2019. Variational adversarial active learning. In *Proc. of the IEEE/CVF International Conference on Computer Vision*. IEEE, Seoul, Korea (South), 5972–5981.
- [12] Francesco Tonini, Cigdem Beyan, and Elisa Ricci. 2022. Multimodal Across Domains Gaze Target Detection. In *Proc. of the 2022 International Conference on Multimodal Interaction*. ACM, Bengaluru, India, 420–431.
- [13] Francesco Tonini, Nicola Dall’Asen, Cigdem Beyan, and Elisa Ricci. 2023. Object-aware Gaze Target Detection. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Paris, France, 21860–21869.
- [14] Danyang Tu, Xiongkuo Min, Huiyu Duan, Guodong Guo, Guangtao Zhai, and Wei Shen. 2022. End-to-end human-gaze-target detection with transformers. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, New Orleans, LA, USA, 2192–2200.
- [15] Donggeun Yoo and In So Kweon. 2019. Learning loss for active learning. In *Proc. of the IEEE/CVF conference on computer vision and pattern recognition*. Computer Vision Foundation / IEEE, Long Beach, CA, USA, 93–102.