Supplementary Material for
Gazeformer: Scalable, Effective and Fast Prediction of Goal-Directed Human Attention

Sounak Mondal[1], Zhibo Yang[1,2], Seoyoung Ahn[1], Dimitris Samaras[1], Gregory Zelinsky[1], Minh Hoai[1,3]
[1]Stony Brook University, [2]Waymo LLC, [3]VinAI Research

In this document, we provide additional experiments, visualizations, details and insights regarding our Gazeformer model. The specific sections of this document are listed below.

- We investigate the impact of layer depth and hidden size on model performance (Section 1).
- We discuss a few details about the components of Gazeformer and its training procedure (Section 2).
- We discuss the performance of Gazeformer and other baseline methods under the ZeroGaze setting for target-absent trials (Section 3).
- We present comprehensive results for model performances under target-absent settings (Section 4).
- We report MultiMatch sub-scores for multiple experiments discussed in the main text (Section 5).
- We present some qualitative examples showcasing Gazeformer's ability to extend to unknown categories (Section 6).
- We report a comparative analysis of Gazeformer and the baseline methods for the individual COCO-Search18 categories, both grouped by ZeroGaze and GazeTrain settings (Section 7).
- We show visualizations of the attention maps of our model to explain the contextual effect on visual search (Section 8).
- We also present visualizations of scanpaths and attention maps of our model in both ZeroGaze and GazeTrain settings (Section 9).
- We showcase comparative qualitative results for Gazeformer and baseline models under the multiple settings discussed in the main material (Section 10).

1. Impact of Layer Depth and Hidden Size Hyperparameters

Model Configurations			Metrics		
N_{enc}	N_{dec}	d	SS↑	FED↓	NSS↑
6	6	512	**0.504**	**2.072**	**8.375**
3↓	3↓	512	0.503	2.078	8.359
6	6	256↓	0.491	2.167	8.182

Table 1. Performance comparison of the Gazeformer model under the traditional GazeTrain setting for different network depths and hidden size (d). N_{enc} denotes the number of encoder layers, N_{dec} denotes the number of decoder layers. The top row shows the original model configuration ($N_{enc} = 6$, $N_{dec} = 6$, $d = 512$) reported in the main text. ↓ denotes the reduction in the depth or hidden sizes with respect to the original configuration.

Table 1 shows the results of two ablations under the GazeTrain setting. We reduce the number of layers for both encoder and decoder blocks from 6 to 3, and in another ablation reduce the hidden size d (dimensionality of hidden states in the transformer layers) from 512 to 256. We observe that reducing the number of encoder/decoder layers or their hidden size (d) does not considerably change the performance metrics. That being said, we see that higher hidden size d is more important than layer depth for this task. We choose 6 encoder layers and 6 decoder layers with hidden size (d) 512 to be our Gazeformer model configuration of choice because it registers slightly superior performance.

2. Additional Architectural and Training Details

Here we present a few additional details about Gazeformer's architecture and training that were not mentioned in the main text due to space constraints.

2.1. Image Encoder Backbone

We extract the 2048-dimensional image features from the last convolutional layer of the ResNet-50 [4] backbone (before the average pooling layer).

2.2. Transformer Encoder

The transformer encoder consists of N_{enc} stacked standard *transformer encoder layers* [7]. Each encoder layer adds a fixed positional embedding as in [1] to the input vector, followed by the application of self-attention, layer normalization and two consecutive linear transformations to obtain a tensor with the same shape as the input tensor. Note that we apply a ReLU activation function to the first linear transformation output.

2.3. Transformer Decoder

The transformer decoder consists of N_{dec} stacked *transformer decoder layers* [7]. Each decoder layer accepts L *fixation queries* as an $L \times d$ tensor along with the encoder output. It then applies self-attention, layer normalization, and encoder-decoder cross-attention and two consecutive linear transformations to obtain a tensor with the same shape as the fixation queries tensor. The self-attention layer ensures that the fixation embeddings at every decoder layer attend to each other while the cross-attention layer is responsible for computing attention scores between each fixation time step and the patch encodings from transformer encoder. Note that we apply a ReLU activation function to the first linear transformation output.

2.4. Disjoint Optimization

The data collected for scanpath prediction task for visual search is unique since *multiple* trajectories are collected for the *same* image with *different* targets and subjects. Specifically, the training set of COCO-Search18 target-present dataset has 1934 images, but 21622 unique trajectories over various target objects and human subjects. Since the transformer encoder block only sees image data whereas the decoder and rest of the downstream network sees image and task data for various subjects, the encoder may overfit while the rest underfits. We also observe that fixation duration predictors and token validity predictor easily overfit during the training process. Hence, we optimize the model using three *disjoint* optimizing routines: one with a smaller learning rate called SlowOpt (for the encoder block and token validity predictor), another with moderate learning rate called MidOpt (for fixation duration predictor MLP), and one with a larger learning rate called FastOpt (for the rest of the network). Similar maneuvers can be observed in [5] for a text summarization task. We use three AdamW [6] optimizers for FastOpt, MidOpt and SlowOpt. We assign learning rate of 1e-6 to SlowOpt, 2e-6 to MidOpt and 1e-4 to FastOpt. All optimizers share the same weight decay of 1e-4.

2.5. Training details

We use a 24 GB NVIDIA Quadro RTX 6000 GPU to train the model with default hyperparameters. Gazeformer model variants are trained for a maximum of 200 epochs with batch size of 32. We use a dropout rate of 0.1 for the encoder block and modality-specific transformations, 0.2 for the multimodal fusion and the decoder block, and 0.4 for the seven MLP output layers for fixation prediction. We have not performed extensive hyperparameter search and tuning because of resource constraints and maintained the same hyperparameter across tasks and settings to emphasize scalability and generalizability.

3. ZeroGaze Performance for Target-Absent Search

In this section, we investigate the ZeroGaze generalizability of Gazeformer and other baselines to novel categories in target-absent search task. Similar to ZeroGaze for target-present trials that we discussed in the main text, models were trained on target-absent scanpaths corresponding to 17 of the COCO-Search18 categories and tested on the target-absent

	SS↑		SemSS↑		FED↓		SemFED ↓		MM	CC	NSS
	w/o Dur	w/ Dur	w/o Dur	w/ Dur	w/o Dur	w/ Dur	w/o Dur	w/ Dur	↑	↑	↑
IRL [8]	0.307	-	0.356	-	5.278	-	4.909	-	0.782	0.201	3.724
Chen *et al.* [2]	0.100	0.032	0.134	0.041	5.649	150.243	5.431	149.018	0.721	0.008	0.057
FFM [9]	0.285	-	0.300	-	4.609	-	4.503	-	0.732	0.221	**4.842**
GazeFormer-noDur	**0.362**	-	**0.412**	-	**3.889**	-	**3.622**	-	0.839	**0.322**	4.547
GazeFormer	0.360	**0.348**	0.409	**0.395**	3.807	**14.776**	3.548	**13.895**	**0.840**	0.319	4.539

Table 2. Performance comparison for models trained with target-absent data and tested on target-absent data under the ZeroGaze setting. The best performance for each metric is highlighted in bold.

scanpaths of one left-out category in a cross-validation manner. The results are in Table 2. Similar to ZeroGaze results on target-present data, Gazeformer outperforms other baselines significantly in almost all metrics. We also observe that compared to its performance when it has been trained on a test category's target-absent scanpaths (see Table 2 in main text), Gazeformer's performance decreases by only 4% when it has not been trained on that category's target-absent scanpaths. This is considerably less when compared to what we observed in the case of target-present data (26% decrease between GazeTrain and ZeroGaze performance). We posit that this is because in target-absent scenario there is less target guidance (less target-related features in the image) and human eye-movements become more explorative and free-viewing like, which makes the generalization to predicting fixations searching for a new target category less hard.

4. Target-Absent Performance Comparisons

	SS↑		SemSS↑		FED↓		SemFED ↓		MM	CC	NSS
	w/o Dur	w/ Dur	w/o Dur	w/ Dur	w/o Dur	w/ Dur	w/o Dur	w/ Dur	↑	↑	↑
Human	0.398	0.369	0.436	0.404	5.418	15.142	3.519	14.428	0.838	0.537	6.547
IRL [8]	0.304	-	0.349	-	5.239	-	4.906	-	0.808	0.263	4.033
Chen *et al.* [2]	0.350	0.330	0.395	0.380	**3.349**	13.728	**3.166**	12.872	0.813	**0.360**	4.356
FFM [9]	0.360	-	0.413	-	3.500	-	3.231	-	0.814	0.310	5.462
GazeFormer-noDur	0.366	-	**0.419**	-	3.492	-	3.246	-	**0.833**	0.334	5.080
GazeFormer	**0.368**	**0.356**	**0.419**	**0.399**	3.417	**13.428**	3.185	**12.708**	0.825	0.341	**5.563**

Table 3. Performance comparison for models trained with target-present data and tested on target-absent data. The best performance for each metric is highlighted in bold. Performance that exceeds human consistency is underlined.

	SS↑		SemSS↑		FED↓		SemFED ↓		MM	CC	NSS
	w/o Dur	w/ Dur	w/o Dur	w/ Dur	w/o Dur	w/ Dur	w/o Dur	w/ Dur	↑	↑	↑
Human	0.398	0.369	0.436	0.404	5.418	15.142	3.519	14.428	0.838	0.537	6.547
IRL [8]	0.323	-	0.378	-	5.18	-	4.753	-	0.805	0.243	3.887
Chen *et al.* [2]	0.345	0.323	0.347	0.335	**3.323**	**13.522**	3.328	13.317	0.799	**0.447**	**7.244**
FFM [9]	0.362	-	0.413	-	3.903	-	3.587	-	0.814	0.289	4.738
GazeFormer-noDur	0.369	-	0.422	-	3.465	-	**3.214**	-	0.830	0.340	4.918
GazeFormer	**0.375**	**0.361**	**0.438**	**0.417**	3.631	14.163	3.312	**13.119**	**0.844**	0.347	5.136

Table 4. Performance comparison for models trained with target-absent data and tested on target-absent data. The best performance for each metric is highlighted in bold. Performance that exceeds human consistency is underlined.

We presented comparative results for performances of Gazeformer and other baselines on target-absent task in Table 2 of main text. Here, we present more comprehensive versions of those model comparisons. Table 3 shows results for models trained on *target-present* data and tested on target-absent data. Table 4 shows results for the same models trained on *target-absent* data and tested on target-absent data. Interestingly, all models, including Gazeformer were able to predict target-absent search fixations when they were trained only with target-present fixations (as well as when training directly on target-absent fixations). This implies that the eye movement pattern used by humans in the presence of a target appears even in the absence of a target to some extent (e.g., searching the counter top to find a microwave), and if the model is trained with sufficient target-present data, it can be generalized to predict target-absent fixations. Among the models compared, Gazeformer achieved the best performance in both settings. We attribute this result to the effectiveness of the transformer encoder to understand and use the scene context. We demonstrate this through attention visualizations in Section 8 and qualitative model comparisons in Section 10.

5. MultiMatch Sub-scores

	MultiMatch				
	shape	direction	length	position	duration
IRL [8]	0.859	0.593	0.847	0.795	-
Chen *et al.* [2]	0.820	0.543	0.785	0.720	0.207
FFM [9]	0.812	0.561	0.777	0.772	-
GazeFormer-noDur	**0.904**	**0.627**	**0.871**	**0.886**	-
GazeFormer	0.897	0.603	0.862	0.885	**0.718**

Table 5. Performance comparison for ZeroGaze setting based on MultiMatch sub-scores.

	MultiMatch				
	shape	direction	length	position	duration
Human	0.903	0.736	0.880	0.910	0.658
IRL [8]	0.889	0.691	0.869	0.881	-
Chen *et al.* [2]	0.888	0.650	0.835	0.906	0.691
FFM [9]	0.875	0.610	0.867	0.879	-
GazeFormer-noDur	0.905	0.721	0.857	**0.914**	-
GazeFormer	**0.906**	**0.730**	**0.859**	0.911	**0.726**

Table 6. Performance comparison for GazeTrain setting based on MultiMatch sub-scores.

	MultiMatch				
	shape	direction	length	position	duration
Human	0.915	0.666	0.906	0.864	0.663
IRL [8]	0.901	**0.642**	0.888	0.802	-
Chen *et al.* [2]	0.903	0.591	0.891	0.865	0.718
FFM [9]	0.896	0.615	0.893	0.850	-
GazeFormer-noDur	**0.926**	0.628	**0.905**	0.871	-
GazeFormer	0.921	0.610	0.898	**0.872**	**0.740**

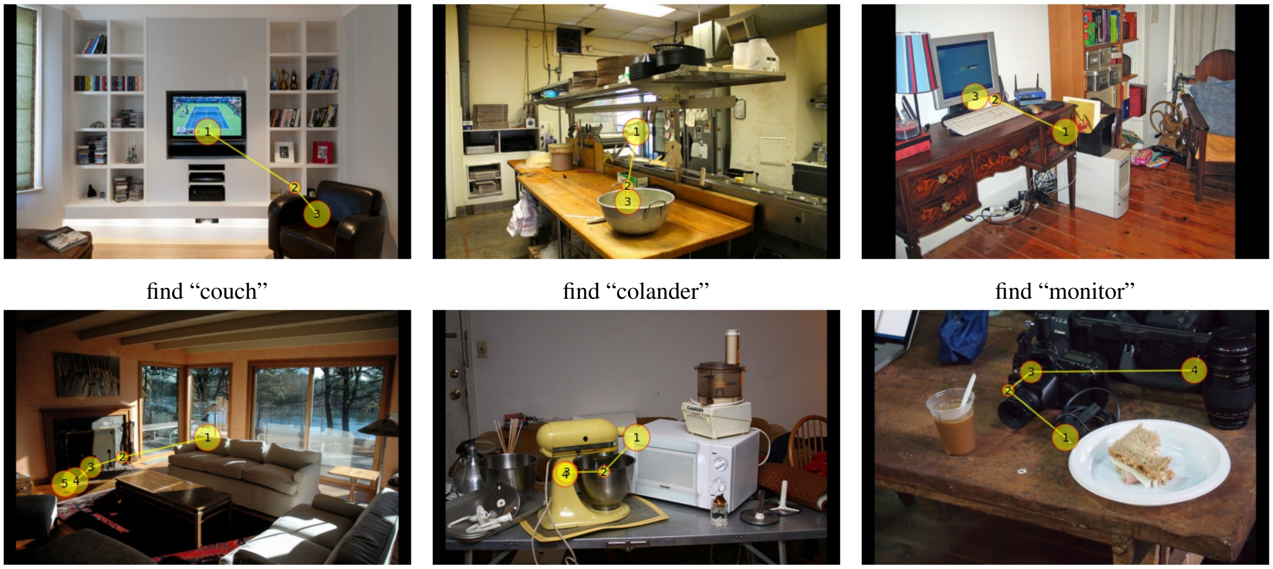
Table 7. Performance comparison for models trained on target-present trials and evaluated on target-absent trials based on MultiMatch sub-scores.

In Table 1 (ZeroGaze and GazeTrain performances) and Table 2 (Target Absent performances) of the main text, we reported only averaged MultiMatch scores for shape, direction, length, position, but the metric consists of 5 different sub-scores. In this section, we present these individual shape, direction, length, position and duration scores for the ZeroGaze setting (Table 1(a) of main text) and the GazeTrain setting (Table 1(b) of main text) in Table 5 and Table 6, respectively. The MultiMatch sub-scores corresponding to Table 2 of main text can be found in Table 7 (here, models are trained on target-present data and evaluated on target-absent data) and Table 8 (here, models are trained on target-absent data and evaluated on target-absent data). Note that IRL [8] and FFM [9] models and GazeFormer-noDur variant do not predict duration. In every table 5, 6, 7, 8, best performances are highlighted in bold and performances that exceed human consistency are underlined.

	MultiMatch				
	shape	direction	length	position	duration
Human	0.915	0.666	0.906	0.864	0.663
IRL [8]	0.885	0.651	0.870	0.815	-
Chen *et al.* [2]	0.892	0.590	0.866	0.848	0.630
FFM [9]	0.891	0.644	0.883	0.838	-
GazeFormer-noDur	0.924	0.625	0.903	0.868	-
GazeFormer	**0.927**	**0.663**	**0.907**	**0.878**	**0.743**

Table 8. Performance comparison for models trained on target-absent trials and evaluated on target-absent trials based on MultiMatch sub-scores.

6. Gazeformer's Extension to Unknown and Unseen Categories


find "couch"

find "colander"

find "monitor"

find "fireplace"

find "stand mixer"

find "camera"

Figure 1. Generalization of Gazeformer to unknown categories. Top row shows extensions to non-canonical names of COCO-Search18's categories; bottom row shows extensions beyond COCO-annotated categories.

In Fig. 1, we showcase more qualitative examples of Gazeformer's ability to extend to multiple categories in-the-wild without use of any prior training data of that category. Gazeformer easily finds COCO-Search18 categories when non-canonical names are used, such as "couch" (instead of "chair"), "colander" (instead of "bowl") and "monitor" (instead of "tv"). Gazeformer's ability to search for novel categories seems even more impressive when asked to search for categories not in the COCO dataset, such as "fireplace", "stand mixer" and "camera". As noted earlier, this ability is nonexistent in previous scanpath prediction models even though it is crucial for use in the real world where a human might use a non-canonical name to refer to *any* arbitrary object for which a detector may not be readily available.

7. Performance Comparison for Individual COCO-Search18 Categories under ZeroGaze and GazeTrain Settings

Fig. 2 compares model performance to human ground truth (using Sequence Score) for each of the 18 target categories in COCO-Search18 under the ZeroGaze setting. Except for three ("car", "stop sign" and "tv"), Gazeformer produced the most human-like scanpaths. Fig. 3 compares model performance to human ground truth (using Sequence Score) for each of the 18 target categories under the GazeTrain setting. Gazeformer particularly performed well for toilets and stop signs, which were

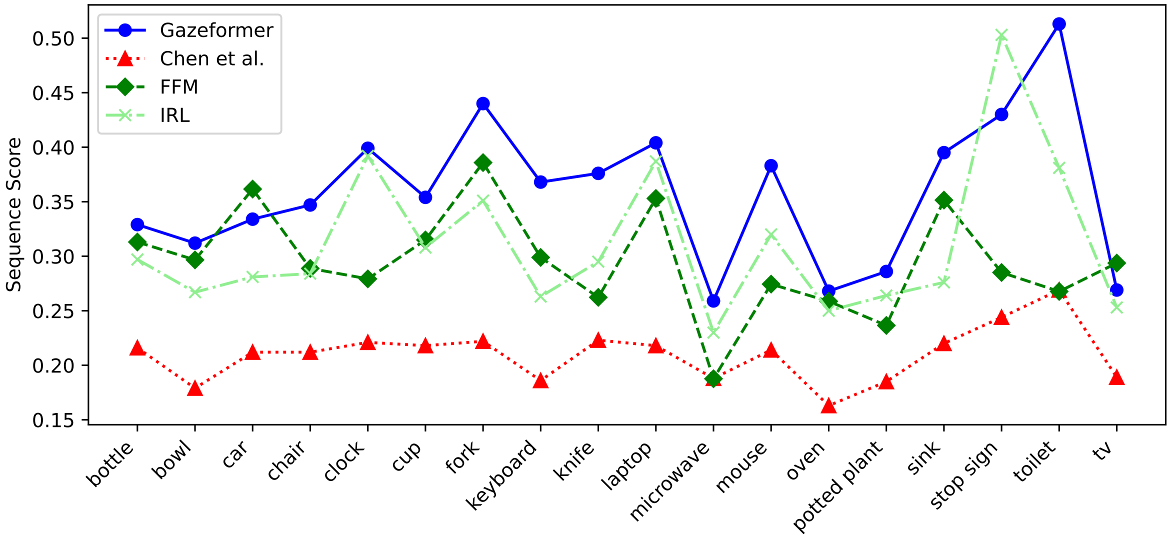


Figure 2. Model performance for each individual category under the ZeroGaze setting

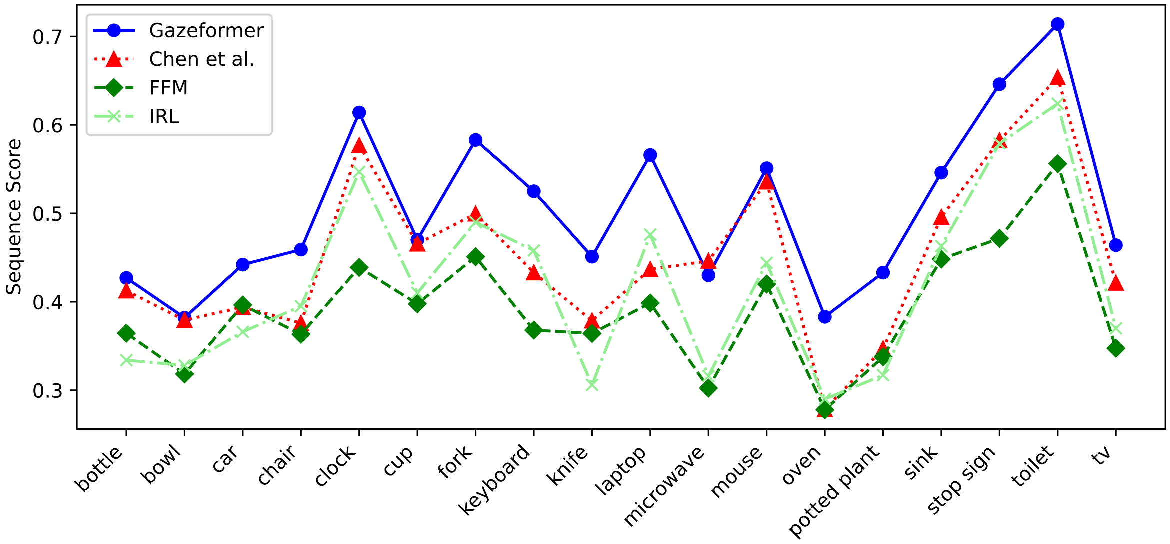


Figure 3. Model performance for each individual category under the GazeTrain setting

among the two easiest COCO-Search18 target categories for humans as well [3].

8. Gazeformer Works Well Because of Context

Why does Gazeformer work so well in predicting search scanpaths? We believe it is largely because of context. The attention mechanism in the transformer layers, together with a semantic embedding of target objects, enables the model to learn a meaningful context of the scene and use it to find targets (e.g., learning that forks can often be found on tables next to plates). One clear test of Gazeformer's use of context is to observe its performance under target-absent search conditions when it is trained on target-present scanpaths. This is because context is all that there is to guide search when the target is absent. This use of context can be seen in visualizations of attention maps, which highlight the image features attended by Gazeformer at every fixation. Attention maps are generated by first averaging the attention weights from the 8 attention heads of the final decoder layer's encoder-decoder cross-attention module, and then expanding these patch-wise attention scores

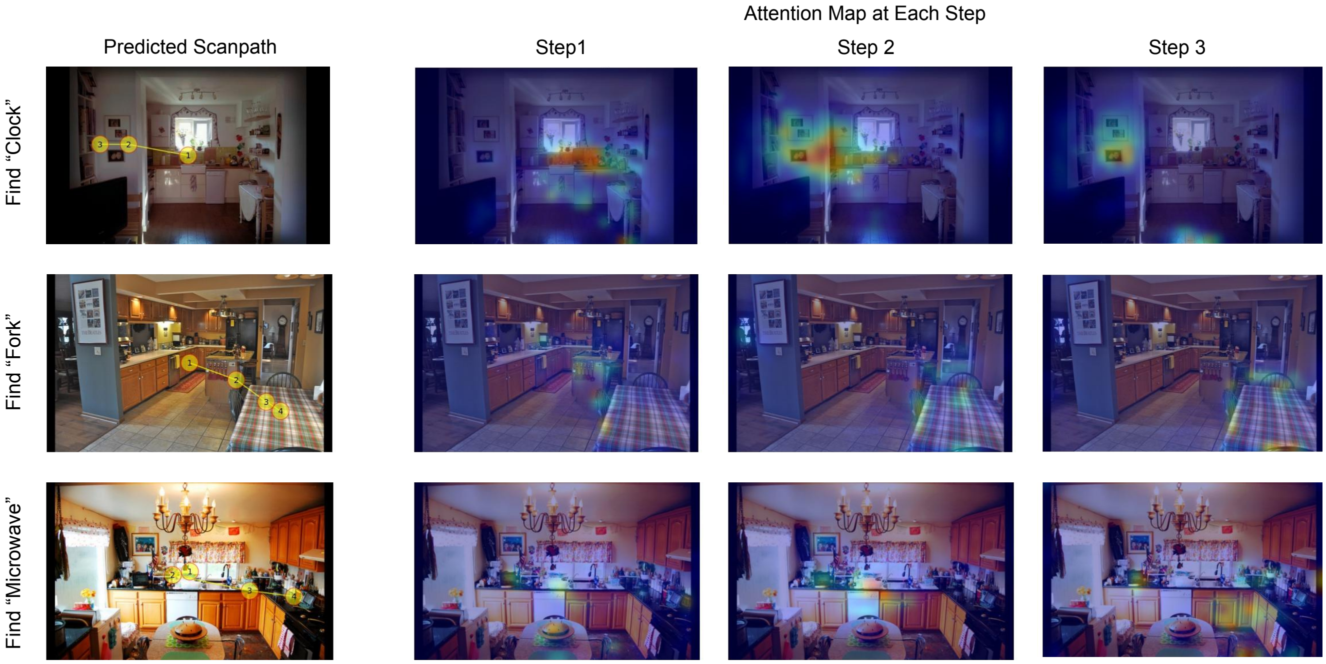
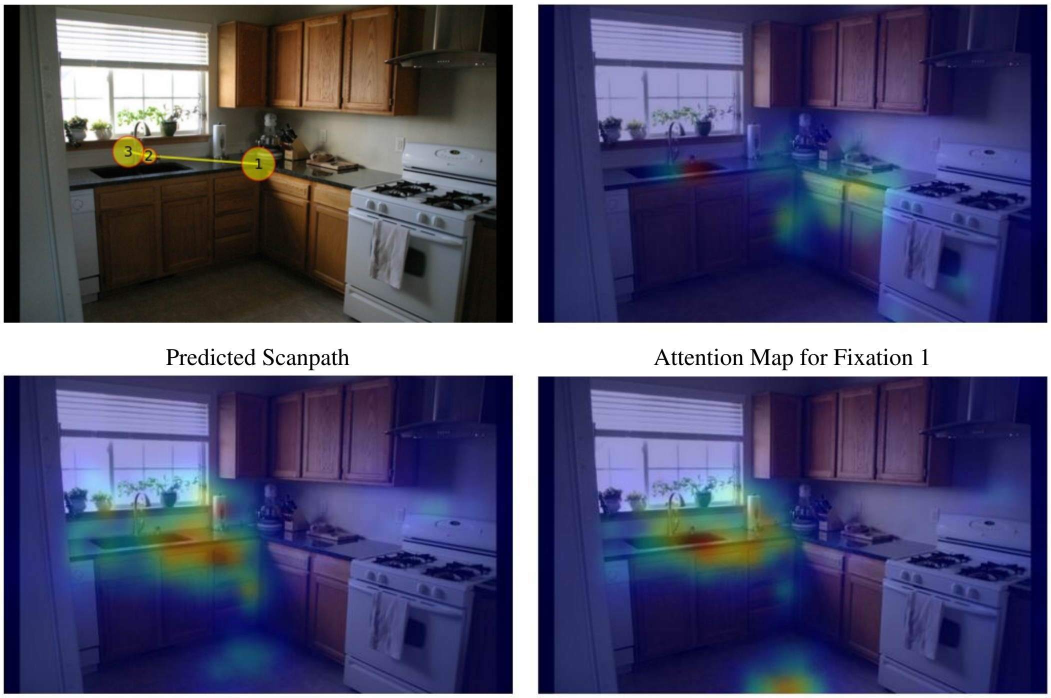


Figure 4. Gazeformer uses scene and object context when there is no target, such as finding a place where the target is most likely to appear, e.g., clock on the wall (top row), fork on the table (middle row), and microwave on the countertop (bottom row). The first column shows scanpaths generated in each case. The remaining columns show attention maps extracted from the final decoder layer of the model (see text for more details). Note that in this scenario, Gazeformer is trained on target-present scanpaths and is asked to predict target-absent scanpaths.

to every pixel in the image and applying Gaussian blur to smooth the maps. Fig. 4 shows representative results. Despite the model seeing a kitchen scene in each example, it looked at different scene areas depending on the target category, e.g., looking for a clock on the wall (top row), a fork on the table (middle row), and a microwave on the counter-top (bottom row)—a clear sign of it using contextual information about object-scene relationships to search for a target. This contextual guidance occurs very early in the process of generating scanpath fixations, indicated in the attention maps by the contextually meaningful regions for each target category attended within the first three fixations.

9. Attention Maps for the GazeTrain and ZeroGaze settings

Here we present scanpaths and corresponding attention maps from Gazeformer for a few test cases in GazeTrain and ZeroGaze settings. The attention maps are collected from the encoder-decoder cross attention at the last decoder layer (see Section 8 for details). Figs. 5, 6, 7 correspond to the GazeTrain setting. Figs. 8, 9, 10 correspond to the ZeroGaze setting.



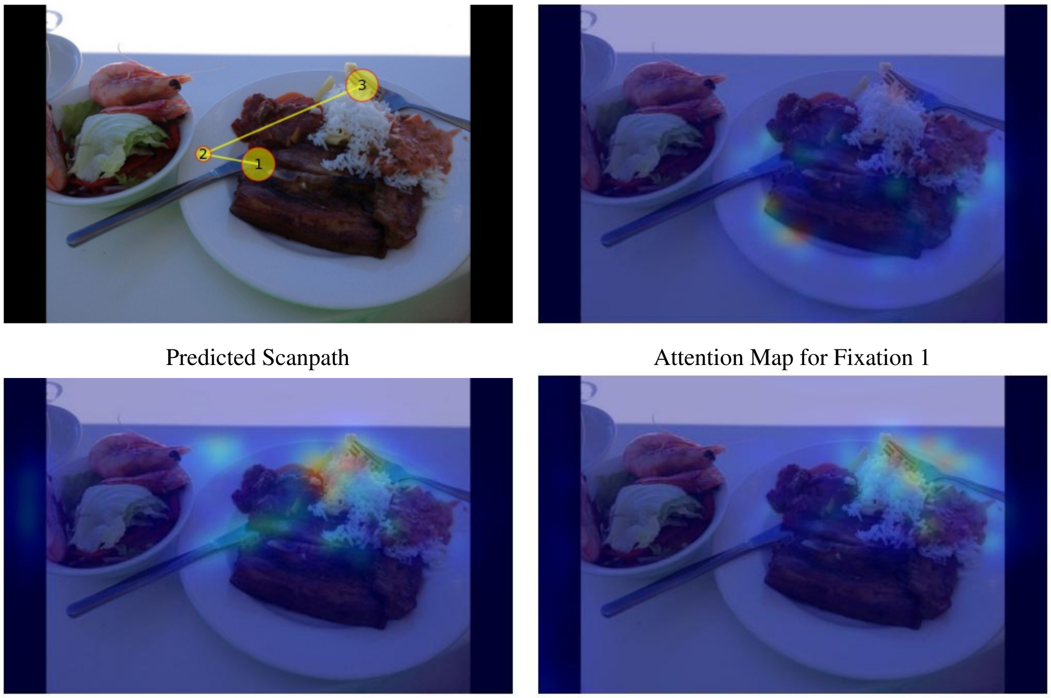
Predicted Scanpath

Attention Map for Fixation 1

Attention Map for Fixation 2

Attention Map for Fixation 3

Figure 5. Predicted scanpath and attention maps for the first three fixations under the **GazeTrain** setting for target **"sink"**. Note that the model attends to the countertop on its first change in fixation, demonstrating its use of context to guide search.



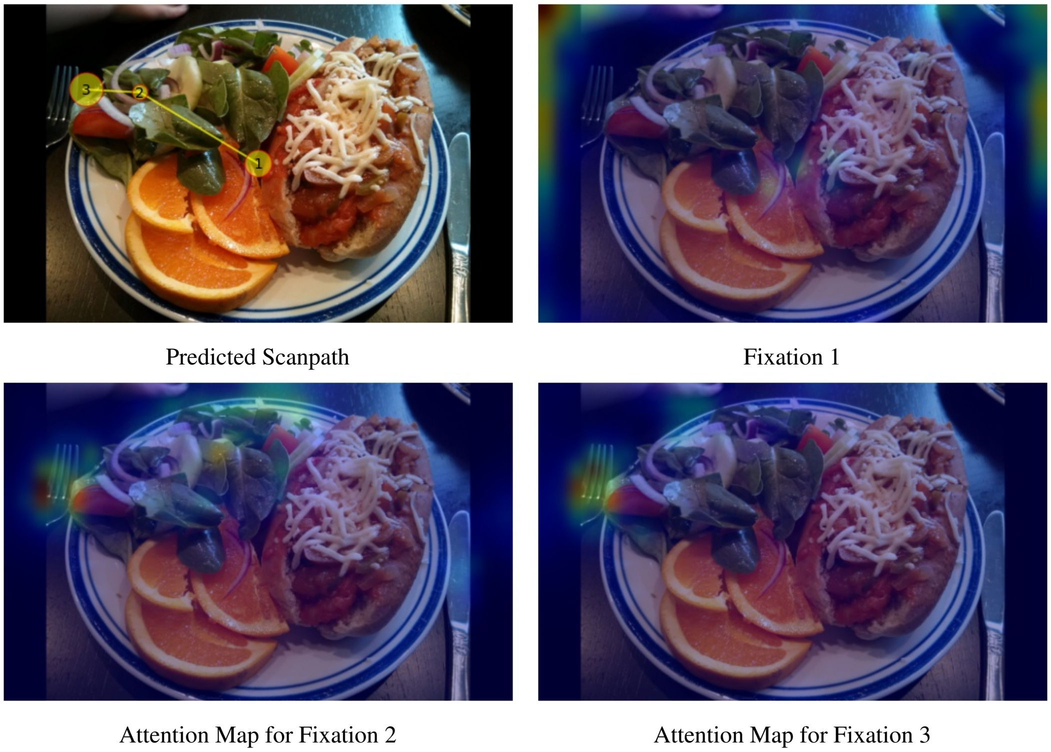
Predicted Scanpath

Attention Map for Fixation 1

Attention Map for Fixation 2

Attention Map for Fixation 3

Figure 6. Predicted scanpath and attention maps for the first three fixations under the **GazeTrain** setting for target **"fork"**. Despite the partial visibility of the fork, Gazeformer manages to find it by first attending to the periphery of the plate.



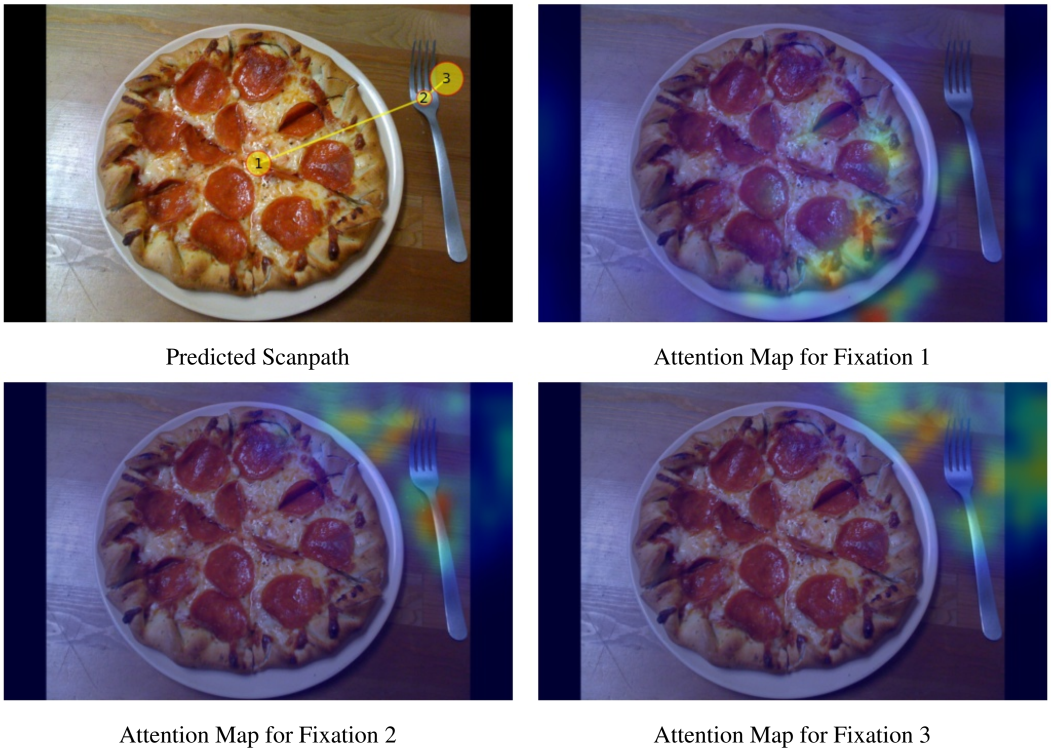
Predicted Scanpath

Fixation 1

Attention Map for Fixation 2

Attention Map for Fixation 3

Figure 7. Predicted scanpath and attention maps for the first three fixations under the **GazeTrain** setting for target **"knife"**. Gazeformer appears to have learned to attend to objects near the periphery of a plate, and in this example was distracted by the fork.



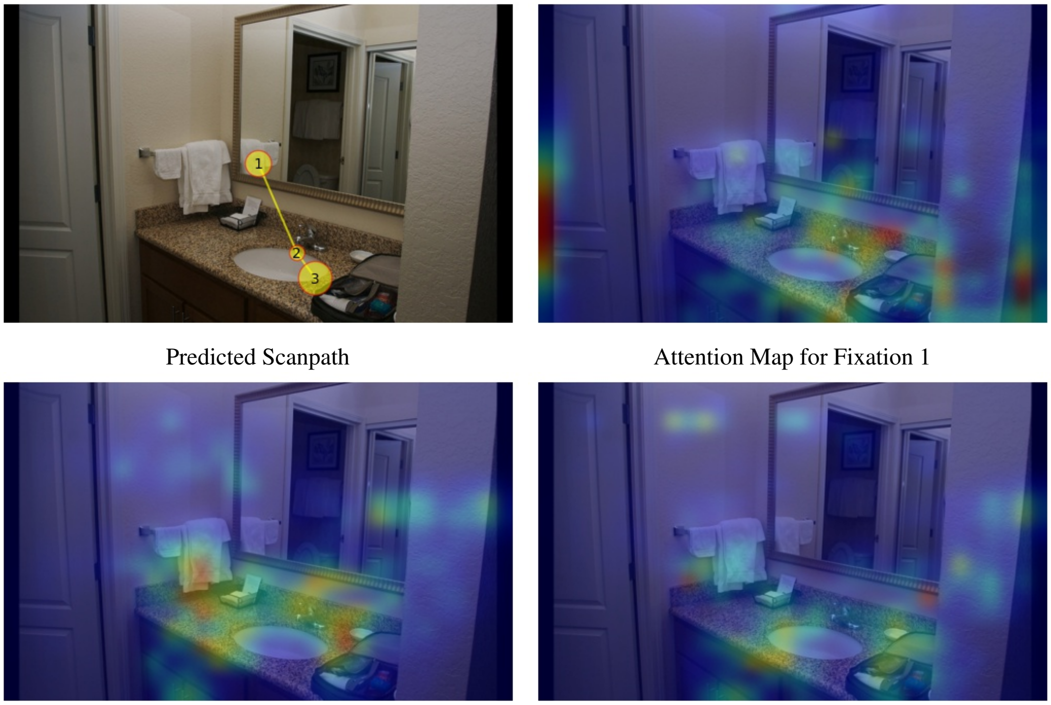
Predicted Scanpath

Attention Map for Fixation 1

Attention Map for Fixation 2

Attention Map for Fixation 3

Figure 8. Predicted scanpath and attention maps for the first three fixations under the **ZeroGaze** setting for target **"fork"**. The attention maps show that on the initial fixation, Gazeformer is attending to the periphery of the plate and the tabletop, where the fork "should be".



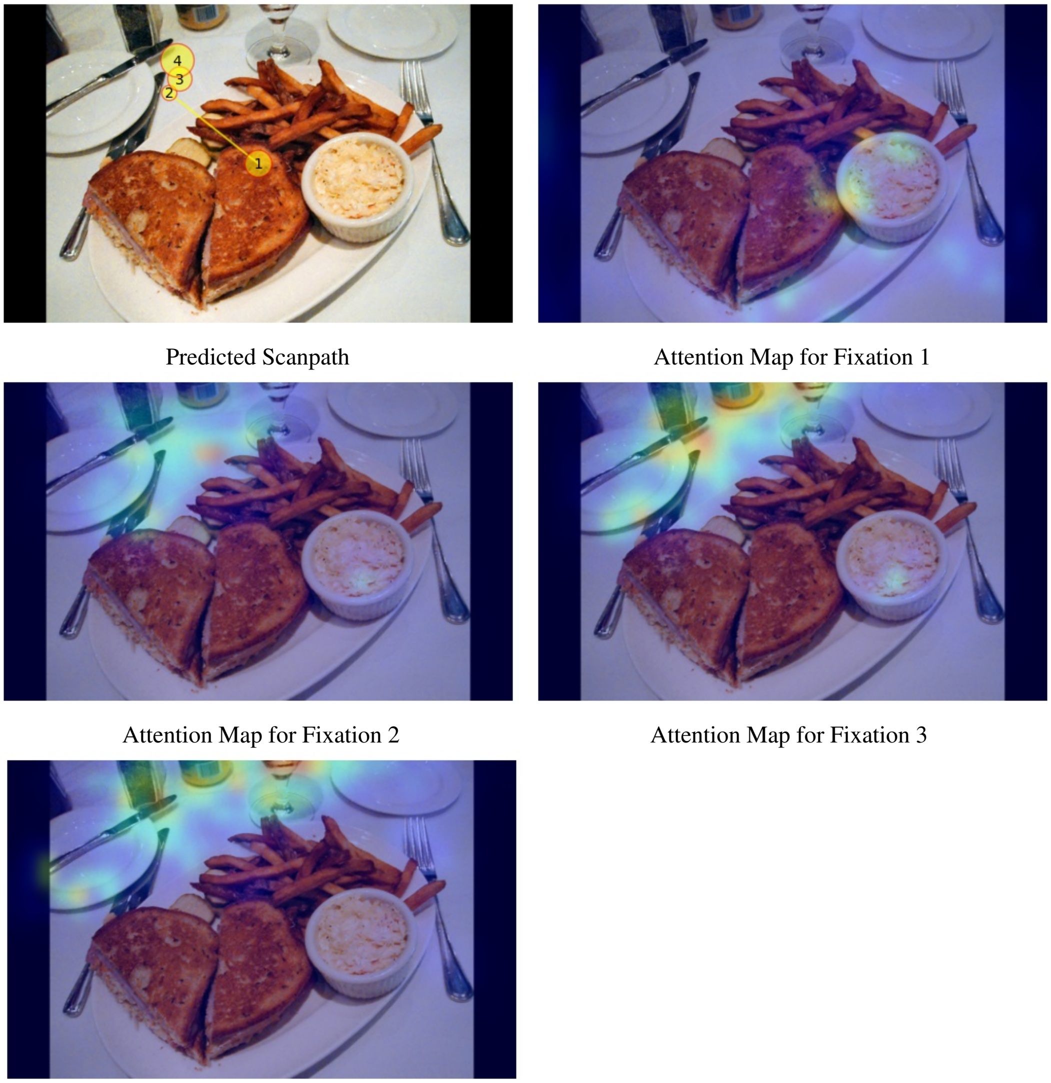
Predicted Scanpath

Attention Map for Fixation 1

Attention Map for Fixation 2

Attention Map for Fixation 3

Figure 9. Predicted scanpath and attention maps for the first three fixations under the **ZeroGaze** setting for target **“sink”**. Gazeformer attends broadly to the countertop where the sink ”should be”, evident in both the scanpath and the attention maps.



Predicted Scanpath

Attention Map for Fixation 1

Attention Map for Fixation 2

Attention Map for Fixation 3

Attention Map for Fixation 4

Figure 10. Predicted scanpath and attention maps for the first four fixations under the **ZeroGaze** setting for target **"fork"**. Knives were particularly distracting for Gazeformer in this setting.

10. Qualitative Comparison

This section contains qualitative results for Gazeformer and other baseline methods on GazeTrain, ZeroGaze and two target-absent settings presented in the main text. We present visualizations for scanpaths corresponding to ZeroGaze setting (Fig. 11), GazeTrain setting (Fig. 12), target-absent evaluation after training on target-present trials (Fig. 13) and target-absent evaluation after training on target-absent trials (Fig. 14).

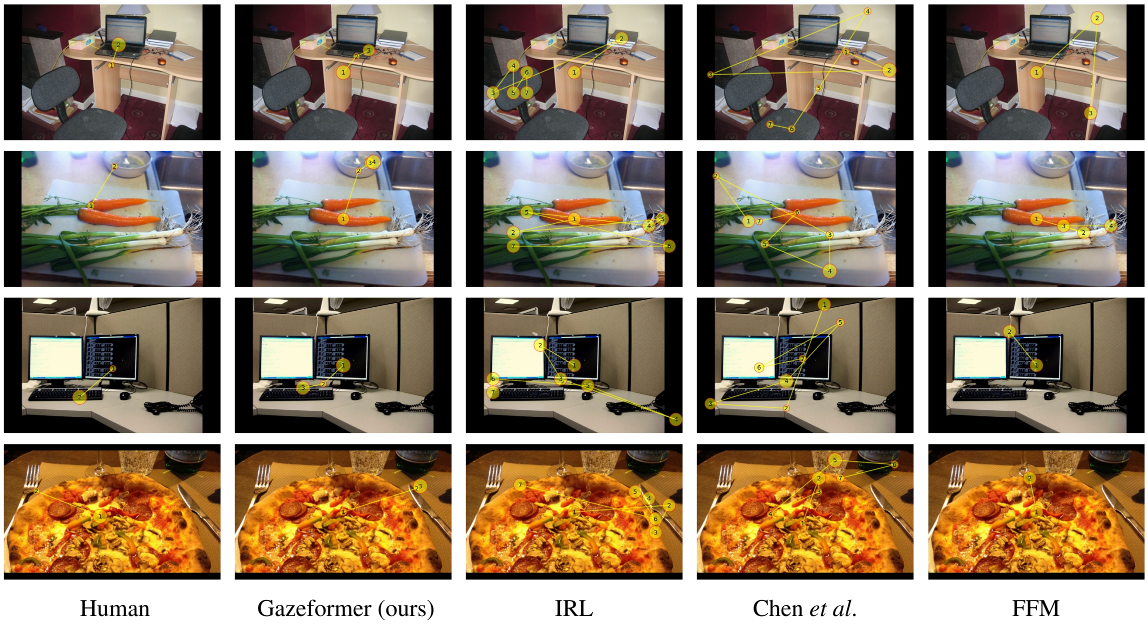


Figure 11. Comparison of scanpath predictions for Gazeformer and baseline models in the *ZeroGaze* setting. Targets in the top three rows are “laptop”, “bowl” and “keyboard”, respectively, with Gazeformer successfully predicting the scanpaths. The bottom row shows a failure case where the knife distracted Gazeformer from the target “fork”.

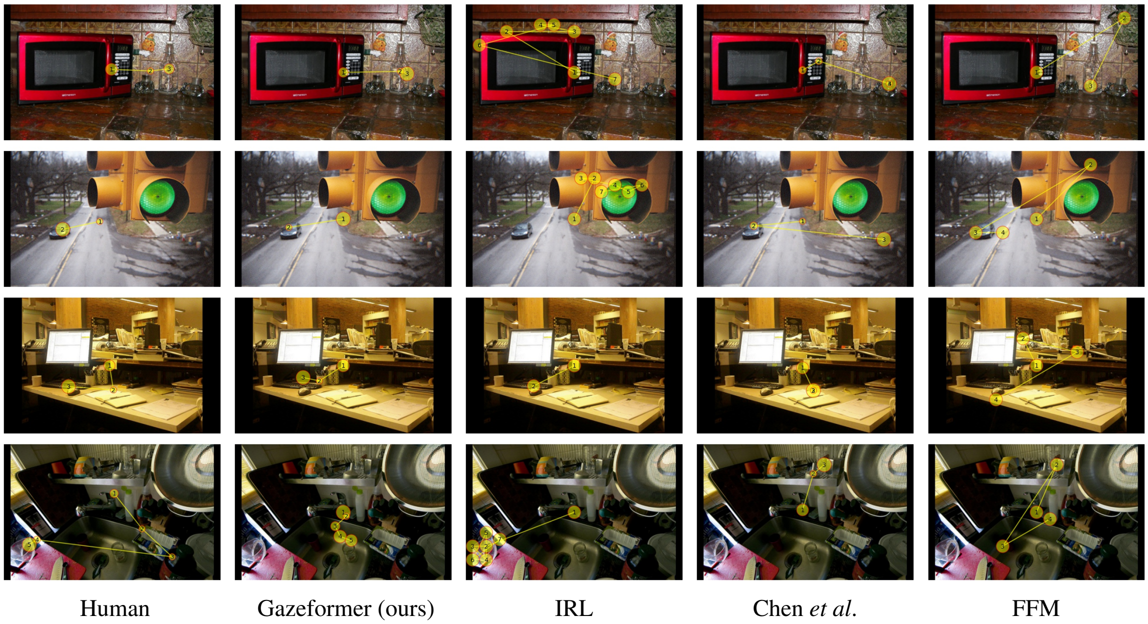


Figure 12. Comparison of scanpath predictions for Gazeformer and baseline models in the *GazeTrain* setting. Targets in the top three rows are "bottle", "car" and "keyboard", respectively, with Gazeformer successfully predicting the scanpaths. The bottom row shows a failure case where the glass jar distracted Gazeformer from the target "cup".

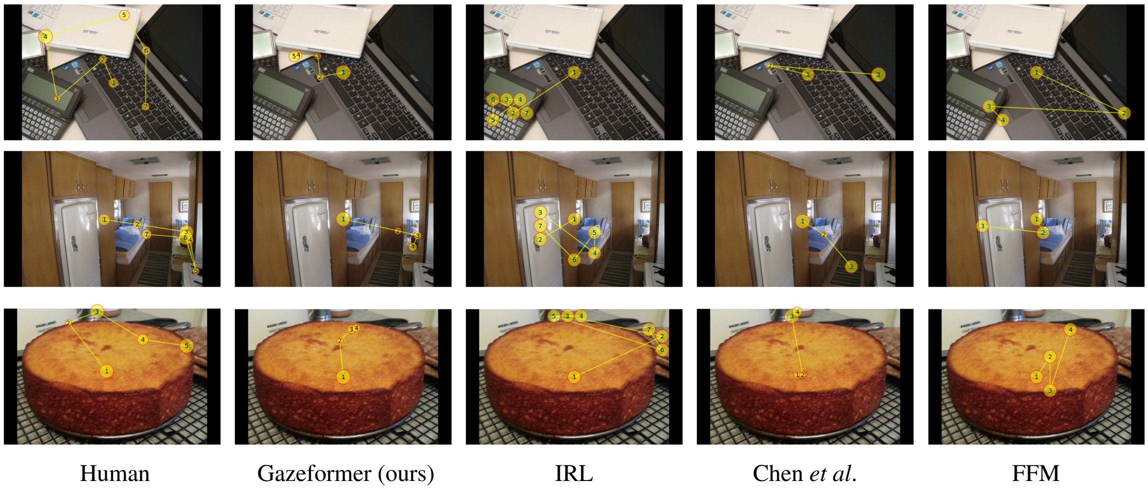


Figure 13. Comparison of scanpath predictions for Gazeformer and baseline models on target-absent trials *when trained only on target-present trials*. Targets in the top two rows are "mouse" and "sink", respectively, with Gazeformer successfully predicting the scanpaths in target-absent setting, exhibiting context effects as it looks for region next to laptop for "mouse" and countertops for "sink". The bottom row shows a failure case where Gazeformer fails to explore outside the confines of the cake while searching for a "bowl".

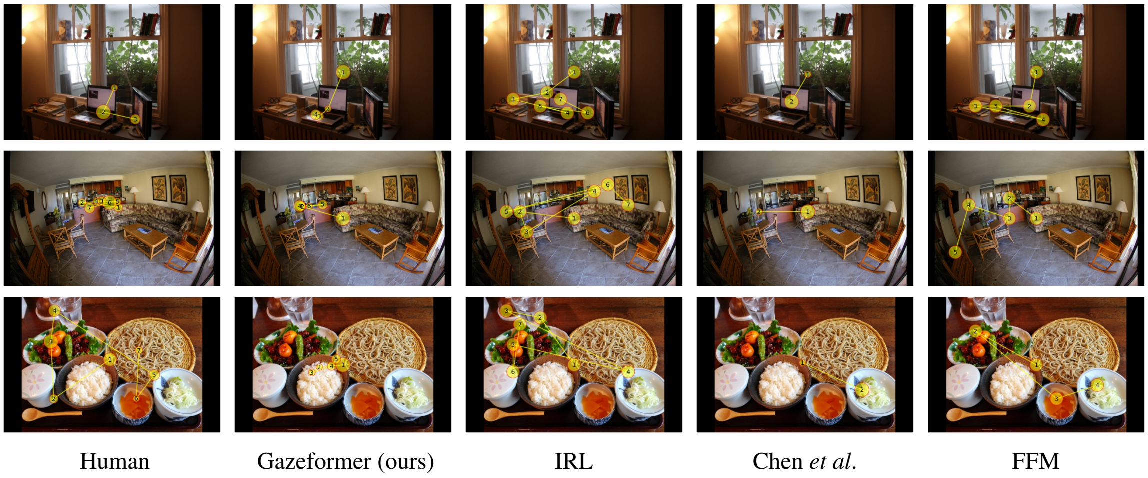


Figure 14. Comparison of scanpath predictions for Gazeformer and baseline models on target-absent trials *when trained only on target-absent trials*. Targets in the top two rows are “keyboard” and “microwave”, respectively, with Gazeformer successfully predicting the scanpaths in target-absent setting, exhibiting context effects as it looks for region next to laptop for “keyboard” and countertops for “microwave”. The bottom row shows a failure case where Gazeformer fails to explore outside the confines of the bowl of rice while searching for a “knife”.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*. Springer, 2020. 2

[2] Xianyu Chen, Ming Jiang, and Qi Zhao. Predicting human scanpaths in visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 3, 4, 5

[3] Yupei Chen, Zhibo Yang, Seoyoung Ahn, Dimitris Samaras, Minh Hoai, and Gregory Zelinsky. Coco-search18 fixation dataset for predicting goal-directed attention control. *Scientific reports*, 11(1):8776, 2021. 6

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 2

[5] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. 2

[6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 2

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, **L** ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2

[8] Zhibo Yang, Lihan Huang, Yupei Chen, Zijun Wei, Seoyoung Ahn, Gregory Zelinsky, Dimitris Samaras, and Minh Hoai. Predicting goal-directed human attention using inverse reinforcement learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020. 3, 4, 5

[9] Zhibo Yang, Sounak Mondal, Seoyoung Ahn, Gregory Zelinsky, Minh Hoai, and Dimitris Samaras. Target-absent human attention. In *European Conference on Computer Vision*, 2022. 3, 4, 5