Supplementary Material: Predicting Goal-directed Human Attention Using Inverse Reinforcement Learning

Zhibo Yang[1], Lihan Huang[1], Yupei Chen[1], Zijun Wei[2], Seoyoung Ahn[1],
Gregory Zelinsky[1], Dimitris Samaras[1], Minh Hoai[1]
[1]Stony Brook University, [2]Adobe Inc.

Abstract

This document provides further details about the COCO-Search18 dataset (Sec. 1), Dynamic Contextual Beliefs (Sec. 2), and implementation (Sec. 3). We also include additional results from experiments and ablation studies, and interpretation (Sec. 4).

1. Details about COCO-Search18 Dataset

Data source: The COCO-Search18 dataset annotates COCO [6] with human gaze fixations made during a standard target-present (TP) or target-absent (TA) search task, where on each trial the search image either depicted the target (TP) or it did not (TA). All of the images were selected from the *trainval* set, and detailed descriptions of TP and TA image selection and gaze collection methods are provided below.

Target present image selection:

In addition to the exclusion criteria described in the main text, we also excluded images in which the target was highly occluded or otherwise difficult to recognize. Specifically, we only selected images in which the cropped target-object patch had a classification confidence $>$.99. To train this classifier, we cropped the target in each image (by bounding box) and used these image patches as positive samples. Same-sized image patches of non-target objects were used as negative samples. Negative samples were constrained to intersect with the target by 25% (area of intersection divided by area of target) so that they could serve as hard negatives for specific targets. More than 1 million cropped patched were collected and resized to 224x224 pixels, while keeping the original aspect ratio by padding. The classifier is fine-tuned from an ImageNet-pretrained ResNet-50 model with the last fully connected layer changed from 1000 outputs to 33 (32+"Negative"). Images with a classification score for the cropped target patch that was $<$.99 were excluded. This resulted in 18 categories with at least 100 images in each category, and 3131 images in total. As described in the main text, we conducted a final manual checking of the dataset to exclude images depicting digital clocks (5 images), so as to make the clock target category specific to analog clocks, and to remove images depicting content that participants might find objectionable. This latter criterion resulting in the exclusion of 30 images, 22 of which were from the toilet category.

After implemented all exclusion criteria, we selected 3101 target-present images from 18 categories: bottle, bowl, car, chair, clock, cup, fork, keyboard, knife, laptop, microwave, mouse, oven, potted plant, sink, stop sign, toilet, tv. See Table 1 for the specific number of images in each category and the average response accuracy (ACC).

Category	TP images	ACC	TA images	ACC
bottle	166	0.84	166	0.92
bowl	141	0.80	141	0.90
car	104	0.89	104	0.91
chair	253	0.89	253	0.64
clock	119	0.99	119	0.97
cup	276	0.92	276	0.76
fork	230	0.96	230	0.98
keyboard	184	0.92	184	0.98
knife	141	0.89	141	0.97
laptop	123	0.95	123	0.95
microwave	156	0.97	156	0.95
mouse	109	0.97	109	0.97
oven	101	0.91	101	0.93
potted plant	154	0.84	154	0.95
sink	279	0.97	279	0.94
stop sigh	126	0.95	126	0.99
toilet	158	0.99	158	1.00
tv	281	0.96	281	0.93
total/mean	3101	0.92	3101	0.92

Table 1: Number of images and response accuracy (ACC) for TP and TA images grouped by target category.

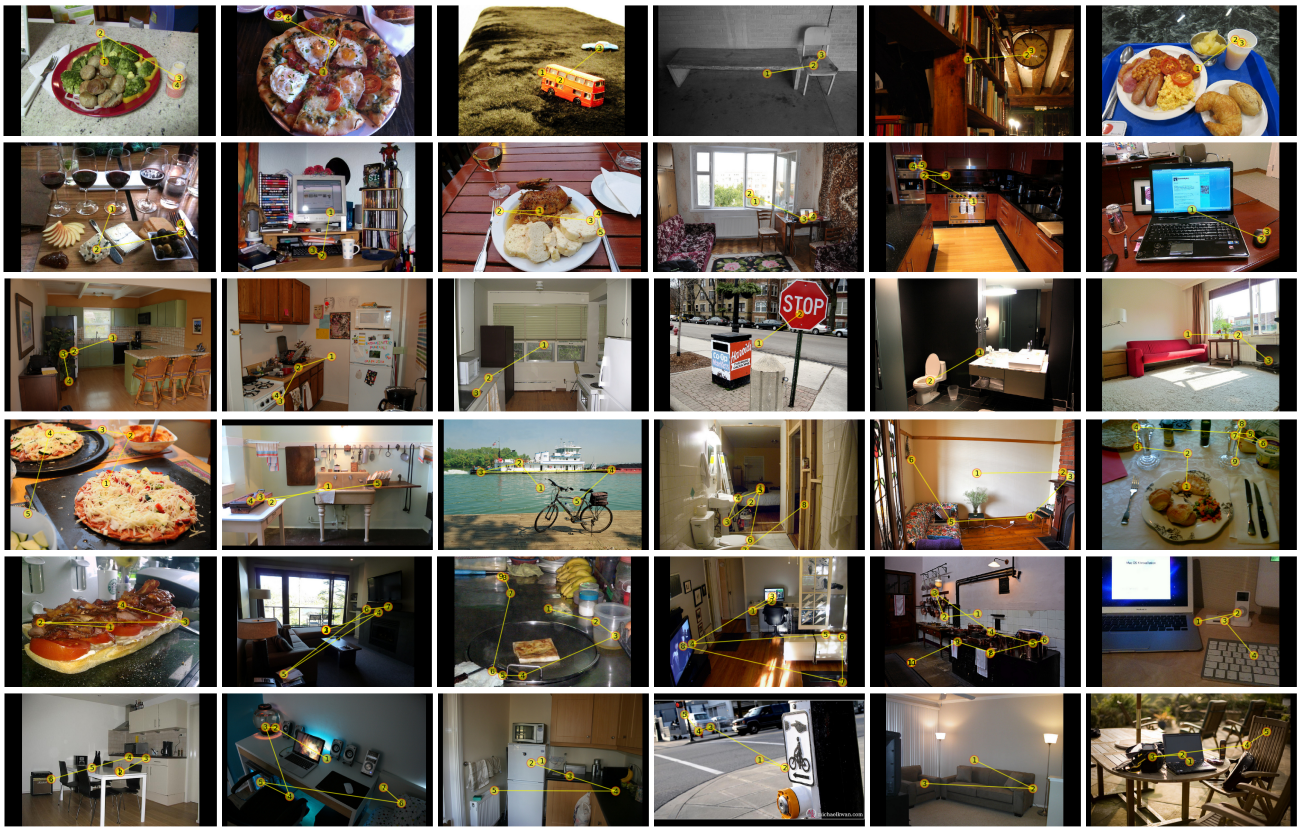

Figure 1: Examples of human scanpaths during target-present (top 3 rows) and target-absent (bottom 3 rows) visual search. From left to right and top to bottom, the 18 target categories are: bottle, bowl, car, chair, clock, cup, fork, keyboard, knife, laptop, microwave, mouse, oven, potted plant, sink, stop sign, toilet, and tv. Each yellow line represents the scanpath of one behavioral searcher, with numbers indicating fixation order.

There were an equal number of TA images (for a total of 6202 images), which were all resized and padded to fit the 1050×1680 resolution of the display monitor.

Gaze data collection procedure: Ten university undergraduate and graduate students (6 males, age range 18–30) with normal or corrected to normal vision participated in this study, which was approved by the Institutional Review Board. They were naive with respect to experimental question and design, and were compensated with course credits or money for their participation. Informed consent was obtained at the beginning of the experiment, and every participant read and understood the consent form before signing it.

The 6202 images were divided into six days of experiment sessions with each session consisting of $\sim$500 TP images and the same number of TA images, randomly interleaved. Images for a given target category were grouped and presented sequentially in an experiment block (i.e., target type was blocked). Preceding each block was a calibration procedure needed to map eye position obtained from the eye-tracker to screen coordinates, and a calibration was not accepted until the average calibration error was $\leq$.51 and the maximal error was $\leq$.94. Each trial began with a fixation dot appearing at the center of the screen. To start a trial, the subject should press the "X" button on a gamepad while carefully looking at the fixation dot. A scene would then be displayed and their task was to answer "yes" or "no" whether an exemplar of the target category for that block appears in the displayed scene. The subject registered a "yes" target judgment by pressing the right rigger of the gamepad, and a "no" judgment by pressing the left trigger. They were told that there were equal number of target present and absent trials, and that they should respond as quickly as possible while remaining accurate. Participants were allowed to take multiple breaks between and within each block.

Image presentation and data collection was controlled by Experiment Builder (SR research Ltd., Ottawa, Ontario, Canada). Images were presented on a 22-inch LCD monitor (resolution: 1050x1680), and subjects viewed these stimuli in a distance of 47cm from the monitor, enforced by both chin rest and head rest. Eye movements were recorded us-

ing an EyeLink 1000 eye tracker in tower-mount configuration (SR research Ltd., Ottawa, Ontario, Canada). The experiment was conducted in a quiet and dimmed laboratory room. Fig. 1 shows some TP and TA images from the 18 object categories, with overlaid human scanpaths.

2. Detailed Description of DCB

DCB: An input image is resized to 320×512 for computational efficiency (the original image is 1050×1680), while the blurred image is obtained by applying a Gaussian filter on the original image with the standard deviation $\sigma = 2$. Both images are passed through a Panoptic-FPN with backbone network ResNet-50-FPN pretrained on COCO2017 [4]. The output of the Panoptic-FPN has 134 feature maps, consisting of 80 "thing" categories (objects) and 54 "stuff" categories (background) in COCO. Feature maps are then resized to 20×32 spatially, same as the discretization of fixation history. At a given time step t, feature maps H for the original image and feature maps L for the blurred image are combined for DCB:

$$B_t = M_t \odot H + (1 - M_t) \odot L \tag{1}$$

where $\odot$ is element-wise product and M_t is the mask generated from fixation history and repeated over feature channels (see Fig. 2). Note that the above equation is equivalent to Eq. (1) in the main paper which is written in a recurrent form.

Encoding the target object category: The task embedding used in our model is the one-hot encoding maps which spatially repeat the one-hot vector. To make predictions conditioned on the task, inputs of each convolutional layer are concatenated with this embedding. This is equivalent to adding a task-dependent bias term for every convolutional layer.

3. Implementation details

Action Space. Our goal is to predict the pixel location where the person is looking in the image during visual search. To reduce the complexity of prediction, we discretize the image into a 20×32 grid, with each patch corresponding to 16×16 pixels in the original image coordinates. This descretized grid defines the action space for all models tested in this paper. At each step, the policy chooses one out of 640 patches and the center location of that selected patch in the original image coordinates is used as an action. The maximum approximation error due to this discretization procedure is 1.75 degrees visual angle.

IRL. The IRL model is composed of three components—the policy network, the critic network and the discriminator network. The **policy network** consists of four convolutional layers whose kernel sizes are 5, 3, 3, 1 with padding 2, 1, 1, 0 and output channels are 128, 64, 32 and 1, and a softmax layer to output a final probability map. The **critic network** has two convolutional layers of kernel size 3 and two fully-connected (fc) layers whose output sizes are 64 and 1. The convolutional layers have output sizes 128 and 256, respectively, and each is followed by a max pooling layer of kernel size 2 to compress the feature maps into a vector. Then this feature vector is regressed to predict the value of the state through two fc layers . The **discriminator network** shares the same structure with the IRL policy network except that the last layer is a sigmoid layer. Note that all convolutional layers and fully-connected layers are followed by a ReLU layer and a batch normalization layer [2] except the output layer.

The critic network is jointly trained with the policy network to estimate the value of a state (i.e., expected return) using smoothed L_1 loss. The estimated value is used to compute the advantage $A(S, a)$ (note that the state S is represented by the proposed DCB in our approach) in Eq. (4) of the main paper using the Generalized Advantage Estimation (GAE) algorithm [8]. At each iteration, the policy network first generates two scanpaths by sampling fixations from the current policy outputs for each image in a batch. Second, we break the generated scanpaths into state-action pairs and sample the same number of state-action pairs from ground-truth human fixations to train the discriminator network which discriminates the generated fixation from behavioral fixations. Lastly, we update the policy and critic network jointly using the PPO algorithm [9] by maximizing the total expected rewards which are given by the discriminator (see Eq. (3) of the main paper).

Training: The IRL model was trained for 20 epochs with an image batch size of 128. The batch sizes used for training the discriminator and policy networks were 64. For the PPO algorithm, the reward discount factor, the clip ratio and number of epochs were set to 0.99, 0.2, and 10, respectively. The extra discount factor in the GAE algorithm was set to 0.96. Both the policy network and the discriminator network were trained with a learning rate of 0.0005. It took approximately 40 minutes to train the proposed IRL model (for 20 epochs) on a single NVIDIA Tesla V100 GPU. The training procedure consumed about 5.6GB GPU memory. Note that the segmentation maps used to construct the DCB state representation had been computed beforehand.

Additional details on two baseline methods. **Detector**: The detector network consists of a feature pyramid network (FPN) for feature extraction (1024 channels) with a ResNet50 pretrained on ImageNet as the backbone and a convolution layer for detection of 18 different targets. The detector network predicts a 2D spatial probability map of the target from the image input and is trained using the

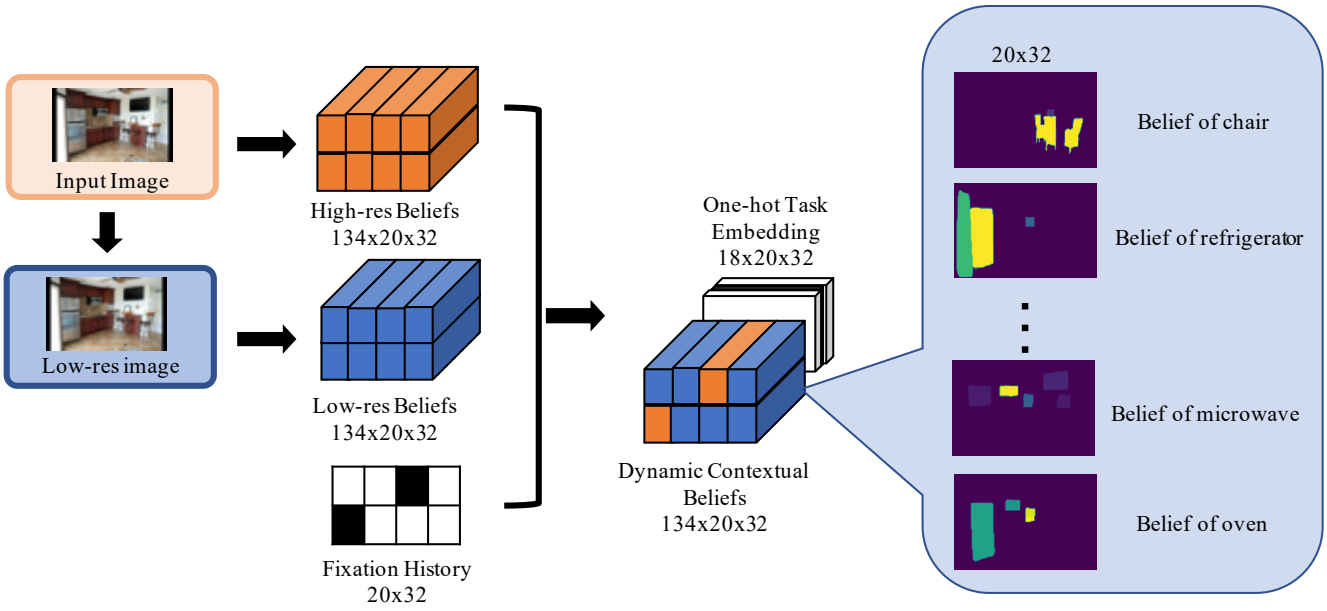

Figure 2: **Detailed illustration of Dynamic-Contextual-Belief**. First, an input image and its low-res image counterpart are converted into high-res beliefs and low-res beliefs. At each fixation, which is discretized into a binary fixation history map with 1's around the fixation location and 0's elsewhere, a new state is generated by concatenating the output of Eq. (1) with a one-hot task embedding (best viewed in color).

ground-truth location of the target. Another similar baseline is **Fixation Heuristics**. This network shares exactly the same network architecture with the detector baseline but it is trained with behavioral fixations in the form of spatial fixation density map (FDM), which is generated from 10 subjects on the training images.

Scanpath Generation. When generating scanpaths, a fixation location is sampled from the probability map that the models have produced and Inhibition-of-Return is applied to prevent revisiting previously attended locations. All predictive methods including IRL, behavior cloning, and heuristic methods, generate a new spatial probability map at every step, while the predicted probability map is fixed over all steps for the Detector and Fixation Heuristic baselines.

4. Additional Experiment Results

Cumulative distribution of sequence scores. In the main paper we reported the *average* Sequence Score of 0.422 for the scanpaths generated by the IRL model. To put this in perspective, Fig. 3 plots the cumulative distribution of the sequence scores and shows four qualitative examples that have sequence scores of 0.33, 0.40, 0.50, and 0.75, respectively.

Comparing different state representations. To evaluate the benefits of having DCB as the state representation, we compared its predictive performance with the Cumulative Foveated Image (CFI) [11] under the same IRL framework. CFI is created by extracting CNN feature maps on the retina-transformed images which are progressively more blurred based on the distance away from the currently fixated location. On the other hand, the DCB is created by extracting panoptic segmentations [3] on uniform-blur images which are uniformly blurred except around the fixated region (the level of blurriness applied in DCB is close to the middle level in the blur pyramid of CFI [1, 7, 11]). For a fair comparison, we extract features for CFI using the backbone ResNet-50-FPN network from the Panoptic-FPN [3] that was used in DCB. Both DCB and CFI have the same spatial resolution of 20×32. As shown in Tab. 2, the IRL model with DCB achieves significantly higher search efficiency and scanpath similarity than when using CFI as state representation. Specifically, DCB reduces the search gap by approximately 45% and improves the scanpath ratio from 61.9% to 82.6%, much closer to the human behavioral ratio of 86.2%. This result is even more impressive considering the size differences between the policy network used with DCB and CFI: DCB is trained with a smaller policy network, since it is comprised of 134 channels, nearly 8x smaller than CFI of 1024 channels. In our experiment, the policy network with CFI state representation has 29.6M parameters, while the policy network with DCB state representation only has 0.3M parameters. Relatedly, another benefit of having DCB as state representation is that it is memory and operation efficient. Creating DCB requires a smaller computational cost than creating CFI, since there's only a single level of blurriness in DCB

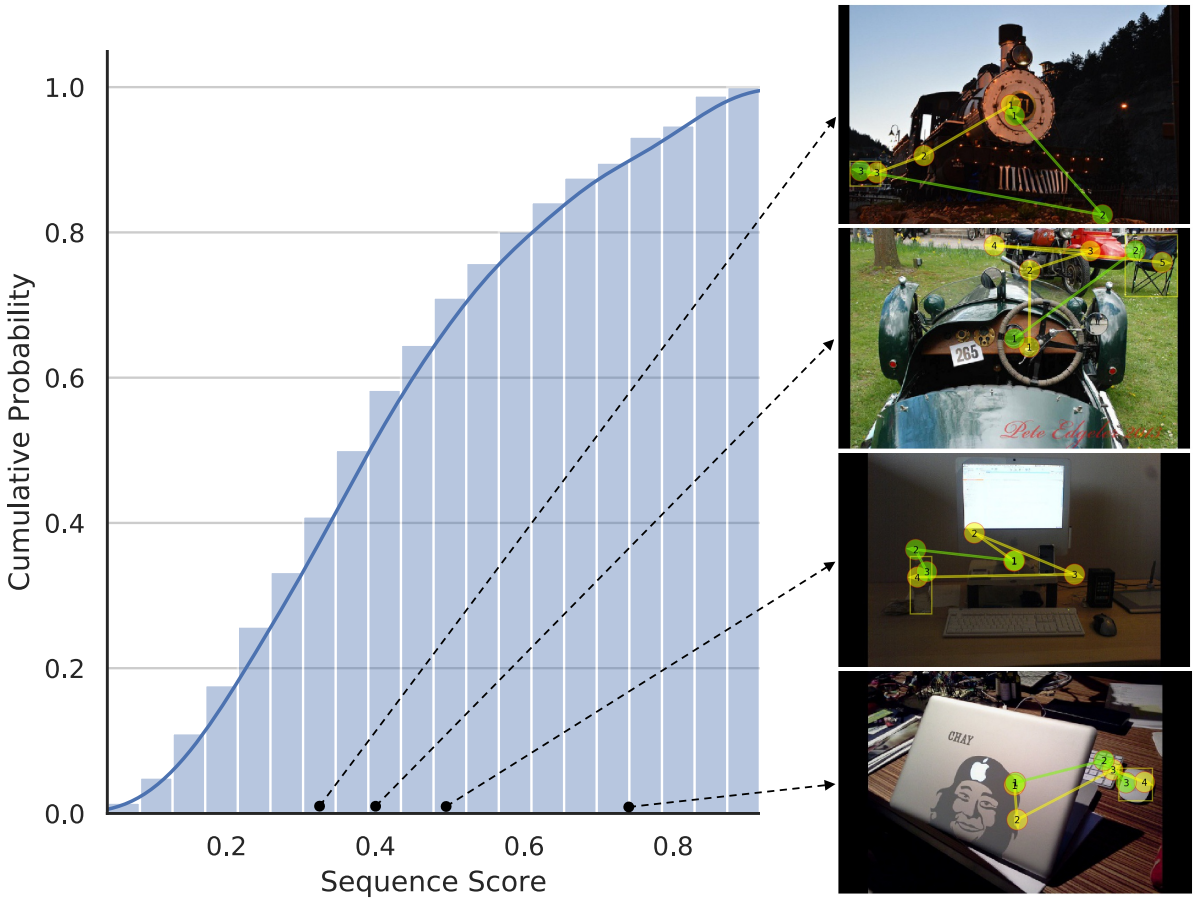

Figure 3: Left: cumulative distribution of the sequence scores of the proposed IRL scanpath prediction method. Right: Four qualitative examples. Human scanpaths are colored in yellow, and the IRL-generated scanpaths are in green. The sequence score for the generated scanpaths are 0.33, 0.40, 0.50, and 0.75, from top to bottom.

and extracted panoptic segmentation maps are smaller by an order of magnitude than the feature maps extracted for CFI. Given that IRL models are particularly difficult to train in high dimensional environments [10], having an efficient representation like DCB can be very helpful.

State Ablation. DCB is a rich representation that incorporates top-down, bottom-up, and history information. The full representation consists of 136 belief maps, which can be divided into five groups: target object (1 map), "thing" (object, 79 maps), "stuff" (background classes, 54 maps), saliency (1 map, extracted using DeepGaze2 [5]), and history (1 binary map for the locations of previous fixations). To understand the contribution of each factor, we removed the maps of each group one at a time and compared the resulting model's performance. As shown in Tab. 3, target object and "thing" maps are the most critical for generating human-like scanpaths, followed by "stuff" maps, whereas saliency and history do not have strong impact to the model performance.

Greedy vs. Non-greedy search behavior. How does human search behavior compare to generated scanpaths reflecting either Greedy or Non-greedy reward policies? Under the greedy policy, the selection of each location to fixate during search reflects a maximization of immediate reward. But the greedy policy is highly short-sighted – it only seeks reward in the short term. Non-greedy reward seeks to maximize the total reward that would be acquired over the sequence of fixations comprising a scanpath. This policy therefore does not maximize reward in the near term, but rather allows more exploration that leads to higher total reward. As shown in Tab. 4, we generated greedy and non-greedy policies from our IRL model and compared their predictive performance on human scanpaths. The results show that 1) models using greedy vs. non-greedy policy produce different search behaviors, with the model using non-greedy policy generating more human-like scanpaths by all tested metrics. This is an interesting finding. Despite the high efficiency of human search in our study (1-2 sec), the search process was strategic in that the fixations maxi-

State Representation	Sequence Score ↑	Scanpath Ratio ↑	TFP-AUC ↑	Probability Mismatch ↓	MultiMatch ↑			
					shape	direction	length	position
DCB	**0.422**	**0.826**	**4.509**	**0.987**	**0.886**	**0.695**	0.866	**0.885**
CFI	0.402	0.619	3.412	1.797	0.875	0.666	0.864	0.857

Table 2: **Dynamic contextual belief (DCB) vs. cumulative foveated image (CFI)** under the framework of IRL.

State Representation	Sequence Score ↑	Scanpath Ratio ↑	TFP-AUC ↑	Probability Mismatch ↓	MultiMatch ↑			
					shape	direction	length	position
DCB with all components	0.422	0.803	4.423	1.029	0.880	0.676	0.841	0.888
w/o history map	0.419	0.800	4.397	1.042	0.882	0.672	0.844	0.887
w/o saliency map	0.419	0.795	4.403	1.029	0.880	0.675	0.840	0.887
w/o stuff maps	0.407	0.777	4.111	1.248	0.876	0.662	0.836	0.875
w/o thing maps	0.331	0.487	2.047	3.152	0.855	0.605	0.852	0.818
w/o target map	0.338	0.519	2.274	2.926	0.866	0.613	0.837	0.820

Table 3: **Ablation study of the proposed state representation—dynamic contextual belief**. The full state consists of 1 history map, 1 saliency map, 54 stuff maps, 79 context maps and 1 target map. We mask out one part by setting the map(s) to zeros at each time.

Scanpath generation policy	Sequence Score ↑	Scanpath Ratio ↑	TFP-AUC ↑	Probability Mismatch ↓	MultiMatch ↑			
					shape	direction	length	position
Based on total reward	**0.422**	**0.826**	**4.509**	**0.987**	**0.886**	**0.695**	0.866	**0.885**
Based on immediate reward	0.375	0.704	3.893	2.143	0.886	0.648	**0.873**	0.852

Table 4: **IRL model predictions using Greedy (immediate reward) and Non-greedy (total reward) policy**.

	Sequence Score ↑	Scanpath Ratio ↑	TFP-AUC ↑	Probability Mismatch ↓	MultiMatch ↑			
					shape	direction	length	position
IRL, 20 ipc	0.415	0.808	4.324	1.140	0.875	0.672	0.832	0.879
CNN, 20 ipc	0.408	0.723	3.906	1.325	0.884	0.664	0.849	0.878
IRL, 10 ipc	0.409	0.774	4.029	1.318	0.881	0.591	0.851	0.819
CNN, 10 ipc	0.397	0.678	3.542	1.657	0.877	0.594	0.847	0.821
IRL, 5 ipc	0.389	0.723	3.696	1.603	0.876	0.588	0.844	0.813
CNN, 5 ipc	0.388	0.678	3.484	1.731	0.886	0.594	0.862	0.828

Table 5: **Data efficiency of IRL and CNN**. "ipc" means images per category used for training. For exmaple, IRL 10 ipc means we train the IRL model using 10 images from each category which are randomly selected from the training data. CNN and IRL are trained and tested on the same images for fair comparison.

mized total reward, even over that short period of time.

Data Efficiency. Table 5 shows the full results of IRL and BC-CNN given different numbers of training images across different metrics. Both use DCB as the state representation. The results are consistent with the results presented in the main paper and suggest that IRL is more data-efficient when compared to the CNN – IRL achieved comparable or better results than the CNN using less training data.

References

[1] Xun Huang, Chengyao Shen, Xavier Boix, and Qi Zhao. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 262–270, 2015. 4

[2] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 3

[3] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019. 4

[4] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 3

[5] Matthias Kummerer, Thomas SA Wallis, Leon A Gatys, and Matthias Bethge. Understanding low-and high-level contri-

butions to fixation prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4789–4798, 2017. 5

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[7] Jeffrey S Perry and Wilson S Geisler. Gaze-contingent real-time simulation of arbitrary visual fields. In *Human vision and electronic imaging VII*, volume 4662, pages 57–70. International Society for Optics and Photonics, 2002. 4

[8] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015. 3

[9] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 3

[10] Aaron Tucker, Adam Gleave, and Stuart Russell. Inverse reinforcement learning for video games. *arXiv preprint arXiv:1810.10593*, 2018. 5

[11] Gregory Zelinsky, Zhibo Yang, Lihan Huang, Yupei Chen, Seoyoung Ahn, Zijun Wei, Hossein Adeli, Dimitris Samaras, and Minh Hoai. Benchmarking gaze prediction for categorical visual search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 4