Supplementary Material: Target-absent Human Attention

Zhibo Yang, Sounak Mondal, Seoyoung Ahn,
Gregory Zelinsky, Minh Hoai, and Dimitris Samaras

Stony Brook University, Stony Brook NY 11794, USA

Abstract. This document provides further details about Foveated Feature Maps (Sec. 1) and network architecture (Sec. 2). We also include additional termination prediction results and compare different backbones (Sec. 3).

1 Detailed description of Foveated Feature Maps

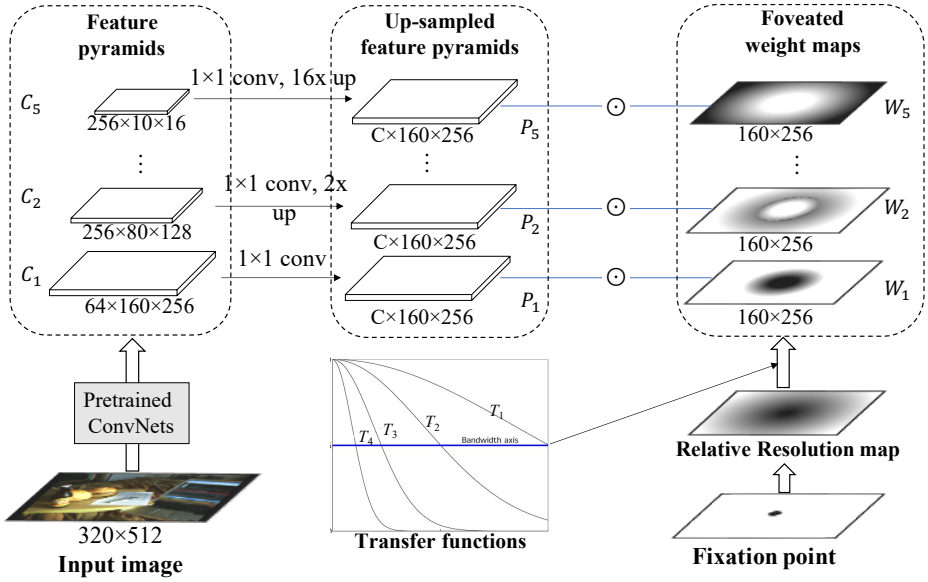

Fig. 1. Foveated feature maps (FFMs) overview. FFMs combine the in-network feature pyramid produced by a pretrained ConvNet based on the previous fixations and a set of predefined transfer functions (see Sec. 3.1 in the main paper). In the foveated weight maps and the relative resolution map, a darker color represents a greater value with all values ranging from 0 to 1.

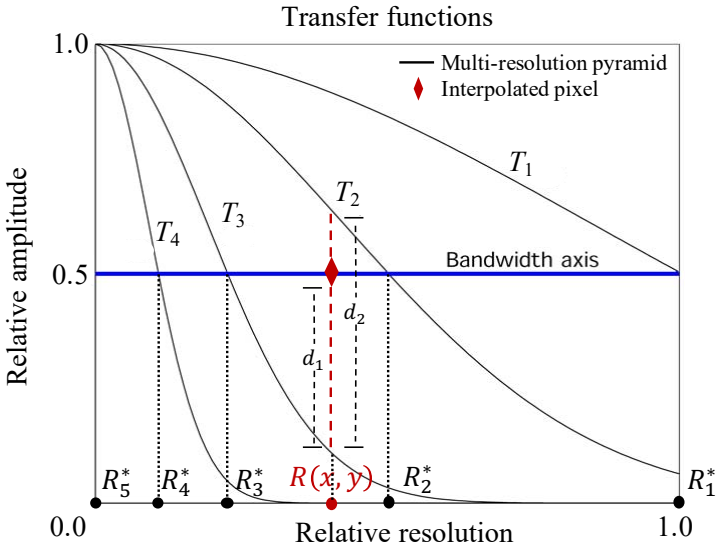

Fig. 2. Transfer functions for the first four levels of the multi-resolution feature pyramid $\{P_1, \cdots, P_4\}$. For example, given a pixel at (x, y) (marked in red) whose relative resolution is $R(x, y)$ and $R_3^* < R(x, y) < R_2^*$, we set the foveated weights at (x, y) at layers other than 2 and 3 to be zero: $W_i(x, y) = 0$ for $i \in \{0, 1, 4, 5\}$. We compute $W_3(x, y)$ as the ratio of the distance between the fixed bandwidth (0.5) and the layer 3 to the distance between the layer 2 and 3 at (x, y) in relative amplitude space: $W_3(x, y) = d_1/d_2$, where $d_1 = 0.5 - T_3(R(x, y))$ and $d_2 = T_2(R(x, y)) - T_3(R(x, y))$.

Fig. 1 gives an overview of the computation of our proposed foveated feature maps (FFMs). Given an input image which is resized to 320×512 and a fixation point (for illustration purposes we only consider a single fixation point), we pass the image input through a pretrained ConvNet (e.g., ResNets [3]) and obtain the feature pyramid denoted as $\{C_1, \cdots, C_5\}$. We project all feature maps to the same depth and upsample them to the spatial size of C_1. We denote the upsampled feature pyramid as $\{P_1, \cdots, P_5\}$. Finally, we compute the final foveated feature maps M as the weighted combination of the upsampled feature maps using a set of foveated weight maps W_i: $M = \sum_i W_i \odot P_i$, where $\odot$ denotes the element-wise multiplication at the spatial axes. W_i is computed based on the relative resolution map $R(x, y) \in [0, 1]$ contingent on the input fixation point (see Eq. (1) of the main paper). Below we describe how to compute W_i based on $R(x, y)$.

To compute the foveated weight maps, we first define a transfer function T_i, which maps relative resolution r to relative amplitude $T_i(r) \in [0, 1]$, for the i-th

level of the pyramid as

$$T_i(r) = \begin{cases} \exp(-(2^{i-3}r/\sigma)^2/2) & i \in \{1, \cdots, 4\} \\ 0 & i = 5 \end{cases} \tag{1}$$

Fig. 2 gives an illustration of the transfer functions and the computation of the foveated weights at (x, y). Each level of the feature pyramid P_i represents a certain eccentricity, corresponding to a fixed spatial resolution, which we denote as R_i^*. R_i^* is defined as the relative resolution where a transfer function $T_i(r)$ is at its half maximum, i.e., $T_i(R_i^*) = 0.5$ [5]. It can be shown that $R_1^* > R_2^* > R_3^* > R_4^* > R_5^* = 0$. We rescale R_i^* such that $R_1^* = 1$. Note that $R_1^*, \cdots, R_5^*$ form four resolution bins whose boundaries are defined by R_i^* and R_{i-1}^* ($i \in \{2, 3, 4, 5\}$). To compute the weights at location (x, y), we first determine which bin pixel (x, y) falls in, according to its relative resolution $R(x, y)$. Assume pixel (x, y) falls in between layer j and $j - 1$, i.e., $R_{j-1}^* \geq R(x, y) > R_j^*$. Then, we compute $W_i(x, y)$ as follows:

$$W_i(x,y) = \begin{cases} \frac{0.5 - T_j(R(x,y))}{T_{j-1}(R(x,y)) - T_j(R(x,y))} & \text{if } i = j-1, \\ 1 - \frac{0.5 - T_j(R(x,y))}{T_{j-1}(R(x,y)) - T_j(R(x,y))} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

It can be seen that $\sum_i W_i(x, y) = 1$ and at location (x, y) only features from layer j and layer $j - 1$ are integrated into the final FFMs.

2 Detailed Network Architecture

Our model has three components: a set of 1×1 convolutional layers that map the feature maps in the feature pyramid to the same dimension (i.e., the number of channels in FFMs); an object detection module; and a fixation prediction module. We set the number of FFMs channels to 128 (i.e., the output channel of the 1×1 convolutional layers). The fixation prediction module and the object detection module share the same ConvNet consisting of three consecutive convolutional blocks which reduce the spatial resolution of the input foveated feature maps (FFMs) by a factor of 8 (from 160×256 to 20×32). Each convolutional block is composed of two convolutional layers whose kernel sizes are 3, 1 with padding 1, 0, output channels are 32, 32 and strides are 1, 2. In between two consecutive convolutional layers of a convolutional block, we apply Layer Normalization [1] and a ReLU activation function. Finally, the fixation prediction module uses two convolutional layers, whose kernel sizes are 3, 1 with padding 1, 0 and output channels are 32 and 18, to map the outputs of the shared ConvNet into 18 attention maps (one for each target in COCO-Search18 [2]). The object detection module has a similar structure, but the output channels of the convolutional layers are 64 and 80 (corresponding to the 80 object categories in COCO), respectively.

Additional details on four baseline methods. **Detector:** The detector network takes the outputs of the "conv4" stage of a pretrained ResNet-50 as input and outputs 18 target object center maps. It consists of two convolutional layers whose kernel sizes are 3, 1 with padding 1, 0, and output channels are 128, 18. Between the convolutional layers, we use batch normalization and ReLU. The detector network predicts a 2D spatial heatmap map of the target from the image input and is trained using the ground-truth location of the target with the target-present images in COCO-Search18. 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 saliency maps (heatmaps indicating how likely each pixel in the images will be fixated by a person), which are generated from the fixations of 10 subjects on the training images. **DCB+IQL-Learn:** The DCB representation [8] is of size $134 \times 20 \times 32$. Following [8], we train a ConvNet with 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. Between two consecutive convolutional layers, we use Layer Normalization and ReLU. **CFI+IQL-Learn:** This network shares exactly the same network architecture with the DCB+IQL-Learn baseline, but it takes the outputs of conv4 of a pretrained ResNet-50 on a cumulative foveated image [9] as input.

3 Additional Experimental Results

3.1 Termination Prediction

Table 1. Effect of different components in termination prediction. The ablated components are: Q-values (640-D), number of fixation (1-D), time (i.e., cumulative fixation duration, 1-D) and subject ID (10-D).

	Q-values	Number of fixations	Time	Subject ID	AuROC
(a)	✓	-	-	-	0.631
(b)	✓	✓	-	-	0.691
(c)	✓	-	✓	-	0.693
(d)	✓	✓	-	✓	0.766

We train a binary classifier to predict the termination of a scanpath. The input to the binary classifier includes 1) *Q-values* (i.e., output from the Q-function) and 2) *number of fixations* (as a rough estimate of time). Here we ablate each component to study the effect of each component in predicting termination. In addition, we ablate two extra components which we did not include in the main

Table 2. Comparing different backbones in FFMs (rows) using multiple scanpath metrics (columns) on the target-absent test set of COCO-Search18. The best results are highlighted in bold.

	SemSS	SS	cIG	cNSS	SS(2)	SS(4)	MAE
ResNet-50	0.516	0.372	**0.729**	**1.524**	0.537	0.441	2.627
ResNet-101	0.510	0.364	0.635	1.465	**0.543**	**0.459**	2.658
VGG16_BN	**0.520**	**0.373**	0.560	1.393	0.531	0.443	**2.570**

paper[1]: 3) *cumulative fixation duration* is the sum of the duration of ground-truth previous fixations (i.e, the time the reviewer has spent in searching for the target); and 4) *subject ID* which is an one-hot vector indicating the subject identity. Here, we ablate cumulative fixation duration to show that the number of scanpath length serves an a good approximation of the time to predict termination; and we ablate subject ID to show that the termination criterion varies significantly across subjects.

Tab. 1 shows the area under the receiver operating characteristic curve (AUROC) of the termination classifier trained with different input combinations. Comparing the first three rows of Tab. 1, we see that "time" plays an important role for predicting scanpath termination and the number of fixations is a good approximation of time (AUROC is only dropped by 0.002 when replacing cumulative fixation duration with the number of fixations). Interestingly, we found that including the subject ID (the last row of Tab. 1) boosted the performance of the termination classifier significantly, which suggests that the termination criterion differs from subject to subject. Some subjects tend to carefully search through the whole image for the target, leading to long scanpaths, whereas some other subjects tend to only scan through the most probable locations and quickly come to a conclusion of whether the image being viewed is target-present or target-absent, leading to scanpaths of much shorter length. This provides additional evidence for our finding in Sec.4.4 of the main paper: individualized modeling may be more suitable for target-absent search prediction.

3.2 Comparing backbones

The proposed FFMs can potentially work with any pretrained ConvNets that are able to produce a feature pyramid. Here, in addition to the results of the main paper which uses ResNet-50 [3] as the backbone, we present the results of our model when combined with other backbones, i.e., ResNet-101 [3] and VGG16 [7] with batch normalization [4] (abbreviated as "VGG16_BN"). Since ResNet-101 has similar structure with ResNet-50, we use the outputs of conv1, conv2, conv3,

[1] We did not include cumulative fixation duration and subject ID in our model of the main paper because our model does not predict fixation duration and is a group model where no subject identity is available, respectively.

conv4, and conv5 in ResNet-101 as the feature pyramid to construct FFMs (see details in Sec. 3.1 of the main paper and Sec. 1 of this supplementary). For VGG16, we use the outputs of each max-pooling layer in VGG16, which has 5 max-pooling layers in total, as the feature pyramid to construct FFMs.

Tab. 2 shows the results of our model with different backbones. It can be seen that ResNet-50, ResNet-101 and VGG16_BN have similar scanpath prediction performance in general. ResNet-50 performs best in cIG, cNSS; VGG16_BN is the best in full-scanpath semantic sequence score (SemSS) and sequences score (SS) due to its best termination prediction performance (i.e., MAE); and ResNet-101 achieves the best sequence scores of fixed-length scanpaths (SS(2) and SS(4)). Notice that the top-1 accuracy of ResNet-50, ResNet-101 and VGG16_BN on ImageNet [6] are 76.1%, 77.4% and 73.4%, respectively. Without considering termination prediction (i.e., performance in fixed-length scanpaths, SS(2) and SS(4)), we see a general trend between the accuracy of the pretrained ConvNets in image recognition and the performance of FFMs in target-absent fixation prediction: the better the backbone performs in image recognition, the better FFM (with that backbone) performs in predicting target-absent fixations.

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
2. Chen, Y., Yang, Z., Ahn, S., Samaras, D., Hoai, M., Zelinsky, G.: Coco-search18 fixation dataset for predicting goal-directed attention control. Scientific reports **11**(1), 1–11 (2021)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
4. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. PMLR (2015)
5. Perry, J.S., Geisler, W.S.: Gaze-contingent real-time simulation of arbitrary visual fields. In: Human vision and electronic imaging VII. vol. 4662, pp. 57–70. International Society for Optics and Photonics (2002)
6. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
7. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
8. Yang, Z., Huang, L., Chen, Y., Wei, Z., Ahn, S., Zelinsky, G., Samaras, D., Hoai, M.: Predicting goal-directed human attention using inverse reinforcement learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 193–202 (2020)
9. Zelinsky, G., Yang, Z., Huang, L., Chen, Y., Ahn, S., Wei, Z., Adeli, H., Samaras, D., Hoai, M.: Benchmarking gaze prediction for categorical visual search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)