Task-driven Webpage Saliency

Quanlong Zheng[1], Jianbo Jiao[1,2],
Ying Cao[1], and Rynson W.H. Lau[1]

[1] Department of Computer Science, City University of Hong Kong
[2] University of Illinois at Urbana-Champaign, Urbana, USA
{qlzheng2-c, jianbjiao2-c}@my.cityu.edu.hk, caoying59@gmail.com,
rynson.lau@cityu.edu.hk

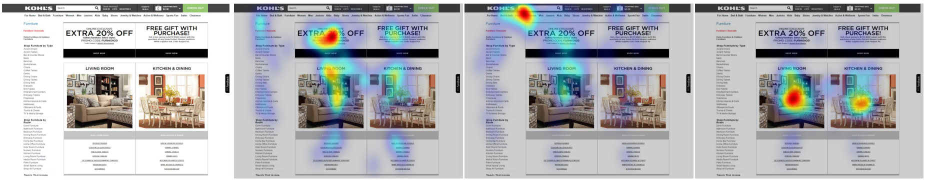
Abstract. In this paper, we present an end-to-end learning framework for predicting task-driven visual saliency on webpages. Given a webpage, we propose a convolutional neural network to predict where people look at it under different task conditions. Inspired by the observation that given a specific task, human attention is strongly correlated with certain semantic components on a webpage (*e.g.,* images, buttons and input boxes), our network explicitly disentangles saliency prediction into two independent sub-tasks: task-specific attention shift prediction and task-free saliency prediction. The task-specific branch estimates task-driven attention shift over a webpage from its semantic components, while the task-free branch infers visual saliency induced by visual features of the webpage. The outputs of the two branches are combined to produce the final prediction. Such a task decomposition framework allows us to efficiently learn our model from a small-scale task-driven saliency dataset with sparse labels (captured under a single task condition). Experimental results show that our method outperforms the baselines and prior works, achieving state-of-the-art performance on a newly collected benchmark dataset for task-driven webpage saliency detection.

Keywords: Webpage analysis · Saliency detection · Task-specific saliency

1 Introduction

Webpages are a ubiquitous and important medium for information communication on the Internet. Webpages are essentially task-driven, created by web designers with particular purposes in mind (*e.g.,* higher click through and conversion rates). When browsing a website, visitors often have tasks to complete, such as finding the information that they need quickly or signing up to an online service. Hence, being able to predict where people will look at a webpage under different task-driven conditions can be practically useful for optimizing web design [5] and informing algorithms for webpage generation [24]. Although some recent works attempt to model human attention on webpages [27,28], or graphic designs [4], they only consider the free-viewing condition.

In this paper, we are interested in predicting task-driven webpage saliency. When visiting a webpage, people often gravitate their attention to different



(a) Input webpage (b) Information browsing (c) Form filling (d) Shopping

Fig. 1: Given an input webpage (a), our model can predict a different saliency map under a different task, *e.g.,* information browsing (b), form filling (c) and shopping (d).

places in different tasks. Hence, given a webpage, we aim to predict the visual saliency under multiple tasks (Fig. 1). There are two main obstacles for this problem: 1) Lack of powerful features for webpage saliency prediction: while existing works have investigated various features for natural images, effective features for graphic designs are ill-studied; 2) Scarcity of data: to our knowledge, the state-of-the art task-driven webpage saliency dataset [24] only contains hundreds of examples, and collecting task-driven saliency data is expensive.

To tackle these challenges, we propose a novel convolutional network architecture, which takes as input a webpage and a task label, and predicts the saliency under the task. Our key observation is that human attention behaviors on webpages under a particular task are mainly driven by the configurations and arrangement of semantic components (*e.g.,* buttons, images and text). For example, in order to register an email account, people tend to first recognize the key components on a webpage and then move their attention towards the sign-up form region composed of several input boxes and a button. Likewise, for online shopping, people are more likely to look at product images accompanied by text descriptions. Inspired by this, we propose to disentangle task-driven saliency prediction into two sub-tasks: task-specific attention shift prediction and task-free saliency prediction. The task-specific branch estimates task-driven global attention shift over the webpage from its semantic components, while the task-free branch predicts visual saliency independent of the task. Our network models the two sub-tasks in an unified architecture and fuses the outputs to make final prediction. We argue that such a task decomposition framework allows efficient network training using only a small-scale task-driven saliency dataset captured under the *single* task condition, *i.e.,* each webpage in the dataset contains the saliency captured on a single task.

To train our model effectively, we first pre-train the task-free subnet on a large-scale natural image saliency dataset and task-specific subnet on synthetic data generated by our proposed data synthesis approach. We then train our network end-to-end on a small-scale task-driven webpage saliency dataset. To evaluate our model, we create a benchmark dataset of 200 webpages, each with visual saliency maps captured under one or more tasks. Our results on this dataset show that our model outperforms the baselines and prior works. Our main contributions are:

- We address webpage saliency prediction under the multi-task condition.
- We propose a learning framework that disentangles the task-driven webpage saliency problem into the task-specific and task-free sub-tasks, which enables the network to be efficiently trained from a small-scale task-driven saliency dataset with sparse annotations.
- We construct a new benchmark dataset for the evaluation of webpage saliency prediction under the multi-task condition.

2 Related Work

2.1 Saliency Detection on Natural Images

Saliency detection on natural images is an active research topic in computer vision. The early works mainly explore various hand-crafted features and feature fusing strategies [1]. Recent works have made significant performance improvements, due to the strong representation power of CNN features. Some works [17, 18, 40] produce high-quality saliency maps using different CNNs to extract multi-scale features. Pan *et al.* [23] propose shallow and deep CNNs for saliency prediction. Wang *et al.* [32] use a multi-stage structure to handle local and global saliency. More recent works [10, 16, 19, 31] apply fully convolutional networks for saliency detection, in order to reduce the number of parameters of the networks and preserve spatial information of internal representations throughout the networks. To get more accurate results, more complex architectures, such as recurrent neural networks [15, 20, 22, 33], hybrid upsampling [38], multi-scale refinement [6], and skip connection [7,9,34]. However, all these works focus on natural images. In contrast, our work focuses on predicting saliency on webpages, which are very different from natural images in visual, structural and semantic characteristics [27].

2.2 Saliency Detection on Webpages

Webpages have well-designed configurations and layouts of semantic components, aiming to direct viewer attention effectively. To address webpage saliency, Shen *et al.* [28] propose a saliency model based on hand-crafted features (face, positional bias, *etc.*) to predict eye fixations on webpages. They later extend [28] to leverage the high-level features from CNNs [27], in addition to the low-level features. However, all these methods assume a free-viewing condition, without considering the effect of tasks upon saliency prediction. Recently, Bylinskii *et al.* [4] propose deep learning based models to predict saliency for data visualization and graphics. They train two separate networks for two types of designs. However, our problem setting is quite different from theirs. Each of their models is specific to a single task associated with their training data, without the ability to control the task condition. In contrast, we aim for a unified, task-conditional framework, where our model will output different saliency maps depending on the given task label.

2.3 Task-driven Visual Saliency

There are several works on analyzing or predicting visual saliency under task-driven conditions. Some previous works [2,12,36] have shown that eye movements are influenced by the given tasks. To predict human attention under a particular task condition (*e.g.,* searching an object in an image), an early work [21] proposes a cognitive model. Recent works attempt to drive saliency prediction using various high-level signals, such as example images [8] and image captions [35]. There is also a line of research on visualizing object-level saliency using image-level supervision [25, 29, 37, 39, 41]. All of the above learning based models are trained on large-scale datasets with dense labels, *i.e.,* each image in the dataset has the ground-truth for all the high-level signals. In contrast, as it is expensive to collect the task-driven webpage saliency data, we especially design our network architecture so that it can be trained efficiently on a small-scale dataset with sparse annotations. Sparse annotations in our context means that each image in our dataset only has ground-truth saliency for a single task, but our goal is to predict saliency under the multiple tasks.

3 Approach

In this section, we describe the proposed approach for task-driven webpage saliency prediction in details. First, we perform a data analysis to understand the relationship between task-specific saliency and semantic components on webpages, which motivates the design of our network and inspires our data synthesis approach. Second, we describe our proposed network that addresses the task-specific and task-free sub-problems in a unified framework. Finally, we introduce a task-driven data synthetic strategy for pre-training our task-specific subnet.

3.1 Task-driven Webpage Saliency Dataset

To train our model, we use a publicly available, state-of-the-art task-driven webpage saliency dataset presented in [24]. This dataset contains 254 webpages, covering 6 common categories: email, file sharing, job searching, product promotion, shopping and social networking. It was collected from an eye tracking experiment, where for each webpage, the eye fixation data of multiple viewers under both a *single* task condition and a free-viewing condition were recorded. Four types of semantic components, **input field**, **text**, **button** and **image** for all the webpages were annotated. To compute a saliency map for a webpage, they aggregated the data gaze data from all the viewers and convolved the result with a Gaussian filter, as in [13]. Note that the size of the dataset is small and we only have saliency data of the webpages captured under the single task condition.

Task definition. In their data collection [24], two general tasks are defined: 1) Comparison: viewers compared a pair of webpages and decided on which one to take for a given purpose (*e.g.,* which website to sign-up for a email service);

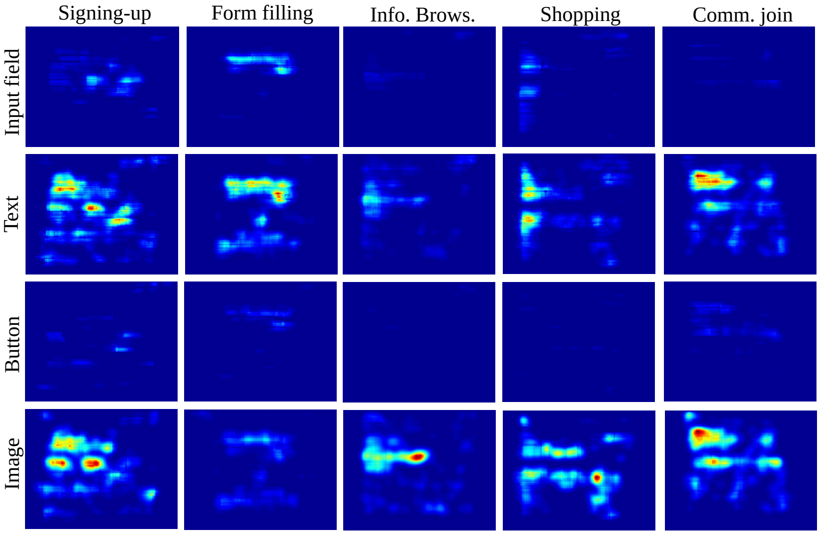


Fig. 2: Accumulative saliency of each semantic component (row) under a specific task (column). From left to right, each column represents the saliency distribution under the Signing-up, Form filling, Information browsing, Shopping or Community joining task. Warm colors represent high saliency. Better view in color.

2) Shopping: viewers were given a certain amount of cash and decided which products to buy in a given shopping website. In our paper, we define 5 common and more specific tasks according to the 6 webpage categories in their dataset: **Signing-up** (email), **Information browsing** (product promotion), **Form filling** (file sharing, job searching), **Shopping** (shopping) and **Community joining** (social networking). We use this task definition throughout the paper.

3.2 Data Analysis

Our hypothesis is that human attention on webpages under the task-driven condition is related to the semantic components of webpages. In other words, with different tasks, human attention may be biased towards different subsets of semantic components, in order to complete their goals efficiently. Here, we explore the relationship between task-driven saliency and semantic components by analyzing the task-driven webpage saliency dataset in Sec. 3.1. Fig. 2 shows the accumulative saliency on each semantic component under different tasks. We can visually inspect some connections between tasks and semantic components. For example, for "Information browsing", the image component receives higher saliency, while other semantic components have relatively lower saliency. Both the input field and button components have higher saliency under "Form filling", relative to other tasks. For "Shopping", both image and text components have higher saliency, while the other two semantic components have quite low saliency.

Table 1: Component saliency ratio for each semantic component (column) under each task (row). The larger the value for a semantic component under a task is, the more likely people look at the semantic component under the task, and vice versa. For each task, we shade two salient semantic components as key components, which are used in our task-driven data synthetic approach.

Task	Input field	Text	Button	Image
Signing-up	0.953	0.971	1.040	1.124
Form filling	1.681	0.979	1.254	0.572
Information browsing	1.725	0.946	0.804	1.033
Shopping	1.444	1.022	0.816	0.770
Community joining	0.895	0.898	1.156	1.186

To understand such a relationship quantitatively, for each semantic component c under a task t, we define a within-task *component saliency ratio*, which measures the average saliency of c under t compared with the average saliency of all the semantic components under t:

$$SR(c,t) = \frac{S_{c,t}}{SA_t}, \tag{1}$$

In particular, $S_{c,t}$ is formulated as: $S_{c,t} = \frac{\sum_{i=1}^{n_{c,t}} s_{c,t,i}}{n_{c,t}}$, where $s_{c,t,i}$ denotes the saliency of the i-th instance of semantic component c (computed as the average saliency value of the pixels within the instance) under task t. $n_{c,t}$ denotes the total number of instances of semantic component c under task t. SA_t is formulated as: $SA_t = \frac{\sum_{c=1}^{n} \sum_{i=1}^{n_{c,t}} s_{c,t,i}}{\sum_{c=1}^{n} n_{c,t}}$, where n denotes the number of semantic components. Our component saliency ratio tells whether a semantic component under a particular task is more salient (> 1), equally salient ($= 1$) or less salient (< 1), as compared with the average saliency. We report the component saliency ratios for all tasks and semantic components in Table 1. We find that, under each task, some semantic components apparently have higher scores than others. This means that people are more likely to look at the high-score semantic components than the low-score ones under the task. For example, for "Form filling", the scores for input and button components are high (1.681, 1.254), while the scores for other semantic components are low (≤ 1), which is consistent with our observation from the accumulative saliency maps above. Based on these component saliency scores, for each task, we identify two semantic components with higher scores as the *key components* (the shaded components in Table 1) that people tend to focus on under the task. These key components are used to synthesize task-driven saliency data for pre-training the task-specific subnet of our network, as introduced in Section 3.5. It is worth noting that when selecting the key components, we also avoid two tasks having exactly the same set of key components, which may confuse the learning of our model. Hence, for "Signing-up", we select "Text" instead of "Button" to prevent "Signing-up" to have the same

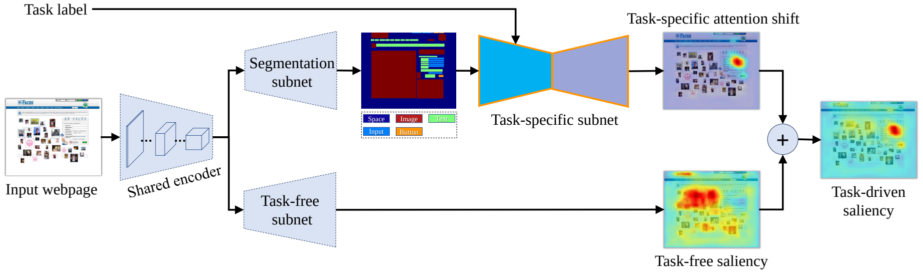


Fig. 3: Network architecture. Inputs to our model are a webpage image and a task label (*e.g.,* "Signing-up"). The webpage image is first fed to a shared encoder to extract high-level features, which are used by two subnets for predicting task-specific human attention bias and task-free visual saliency. The task-specific subnet takes as input the task label along with a semantic segmentation map from a segmentation subnet, and predicts the task-dependent attention shift (upper), while the task-free subnet predicts the task-independent saliency (lower). The task-specific attention shift and task-free saliency are combined to obtain the final saliency map under the input task.

set of key components as "Community joining". The above analysis confirms our assumption that human attention shift under a particular task is correlated with and can thus be predicted from a subset of semantic components.

3.3 Network Architecture

Fig. 3 shows the architecture of our proposed network. A webpage image is first fed into a shared encoder to extract high-level feature representation. The shared encoder uses all the layers of the FCN [26] before the output layer. After that, the network splits into two branches: the task-specific branch and task-free branch. For the task-specific branch, we use a segmentation subnet (using the output layer of the FCN [26]) to generate a semantic segmentation map from the extracted feature representation. We then send a task label (*e.g.,* "Signing-up") along with the semantic segmentation map to a task-specific subnet, which outputs a task-specific attention shift map. For the task-free branch, we use a task-free subnet to map the extracted feature representation to a task-free saliency map. The task-specifc attention shift map and the task-free saliency map are added to produce the final output. We also tried other fusion operations *e.g.,* multiplication, but found addition performs better.

Task-specific subnet: The task-specific subnet is used to model human attention shift towards particular semantic components under the task-driven condition (as validated in Section 3.2). To do this, we first obtain a semantic segmentation map through a segmentation subnet. To account for segmentation uncertainty, we directly take the output of the segmentation layer (probability distributions over different semantic components) as the segmentation map, and then feed

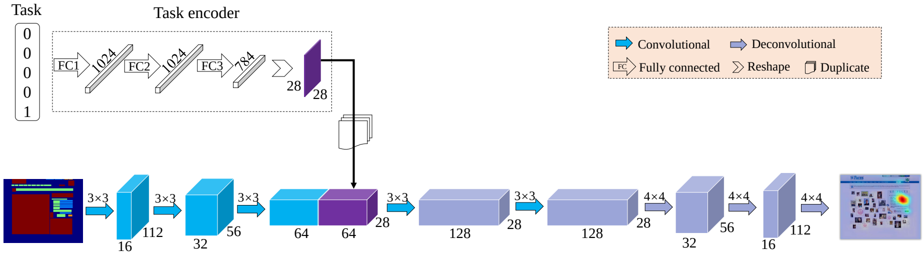


Fig. 4: Task-specific subnet. The filter sizes of the convolutional and deconvolutional layers are labeled above the corresponding layers. The channel numbers and sizes of the feature maps are also labeled nearby the feature maps.

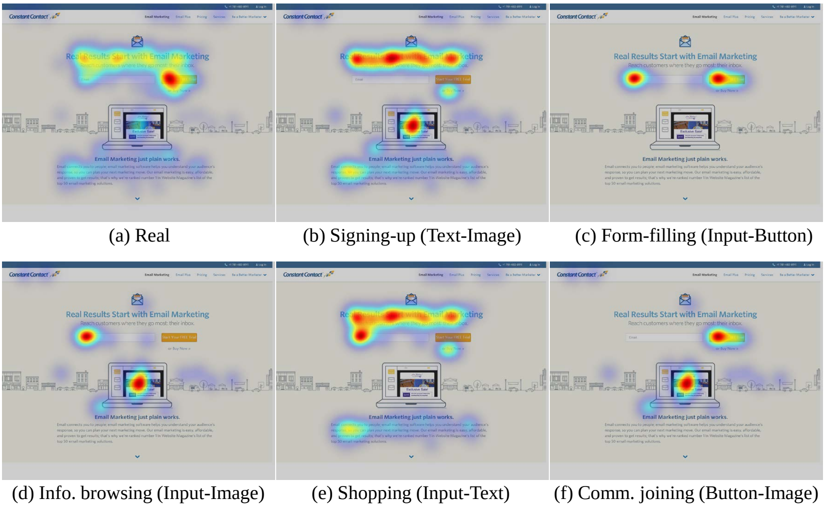
it to the task-specific subnet to predict the attention shift among the semantic components. Fig. 4 shows the detailed structure. The semantic segmentation map is passed through a series of convolutional layers to get a lower-dimensional segmentation representation. To encode the task label, we represent it using one-of-K representation (K=5) and transform it into a semantic vector via a task encoder with a stack of fully connected layers. The semantic vector is then reshaped and duplicated multiple times, and concatenated with the segmentation representation. The concatenated features are finally transformed by a stack of deconvolutional layers to output a task-specific attention shift map.

Task-free subnet: The task-free subnet is used to model visual saliency, which is task-independent and driven by visual contents of the input webpage. To simplify our network, this subnet uses the output layer of the FCN [26] to directly output a saliency map, which works well in our experiments. More complex layers can be added, but at the cost of extra parameters.

Discussion: Our network architecture can be efficiently trained, even with small amounts of training data, to produce reasonable saliency predictions given different tasks. This is because our framework has the task-specific branch to model the task-related saliency shift from task-free saliency. In addition, the task-specific subnet receives a semantic segmentation map, instead of the webpage, as input. The complexity of the input space is greatly reduced, as only several semantic classes need to be encoded. This makes it easier for the model to discover consistent patterns and learn the mapping from a task label to the corresponding attention shift.

3.4 Training

Due to the deep network architecture, directly training it end-to-end on our small dataset is difficult. Thus, we propose a two-stage training strategy, where we first pre-train each part separately and then fine-tune the entire network jointly. In



(a) Real (b) Signing-up (Text-Image) (c) Form-filling (Input-Button)

(d) Info. browsing (Input-Image) (e) Shopping (Input-Text) (f) Comm. joining (Button-Image)

Fig. 5: Synthetic saliency data. (a) Saliency map from the webpage dataset [24]. (b)-(f) Synthesized saliency maps from (a) for 5 different tasks. The corresponding key components of each task are shown in braces.

particular, we first pre-train the task-free subnet on a large-scale natural image saliency dataset, SALICON [11], and then fine-tune it on the webpage saliency dataset [24]. It is trained by minimizing a L2 loss between the predicted and ground-truth saliency, L_{sal}. For the segmentation subnet, we enforce a cross-entry loss between the predicted and ground-truth semantic segmentation maps, L_{seg}, and train it on the webpage saliency dataset with ground-truth semantic annotations. Since the segmentation subnet and task-free subnet share the same encoder, we thus jointly train them with a multi-task loss L_{multi},

$$L_{multi} = L_{sal} + L_{seg}, \tag{2}$$

The task-specific subnet is pre-trained from scratch on a synthetic task-driven saliency dataset (as discussed below), with L2 loss between the predicted and ground-truth attention maps. Finally, we train the entire model end-to-end using L2 loss between the ground-truth and predicted saliency maps given a task label. We have also tried several other loss functions, *e.g.,* cross entropy loss and L1 loss, but found that they produced worse performances.

3.5 Task-driven Data Synthesis

Pre-training the task-specific subnet requires a lot of saliency data on webpages under the multi-task condition, which is not available and expensive to collect. To address this limitation, we propose a data synthesis approach to generate our training dataset by leveraging the key semantic components for each task that we have identified in Section 3.2. Our data synthesis method works as follows. Given a webpage in our dataset, we take its existing task-driven saliency map. For each task of the five tasks, we only preserve the saliency of the saliency map on the corresponding key components of the task, by zeroing out the saliency in other regions. In this way, we generate 5 task-specific saliency maps for each webpage. Fig. 5 shows an example of the synthesized saliency maps under different tasks. With our data synthesis approach, we generate a dataset with dense annotations (*i.e.,* the saliency data under all tasks are available for all webpages), which is sufficient for pre-training our task-specific subnet.

4 Implementation Details

The segmentation subnet and task-free subnet are based on the FCN [26], and we adopt VGG-16 [30] for the shared encoder of the FCN. The parameters are optimized by Adam optimizer [30], with a batch size of 20. During training, we use different learning rates for different parts. For the task-specific and task-free subnets, we set the initial learning rate to be 10^{-7}, and divide it by 10 every 20 epochs. For the shared encoder, we start with a small initial learning rate (10^{-10}) and set it to be the same as that of the task-free subnet after 20 epochs. We train our network for 100 epochs. The webpage images and their saliency maps are resized to 224 × 224.

5 Experiments

In this section, we first introduce the evaluation dataset and evaluation metrics. We then analyze our network architecture and training strategy in an ablation study. Finally, we compare our method with prior methods.

5.1 Evaluation Dataset and Metrics

To evaluate our method, a task-driven webpage saliency dataset is required, where each webpage has ground-truth saliency under different tasks. Unfortunately, such dataset is not available. Thus, we construct a new evaluation dataset, which includes 200 webpages collected from the Internet by us. The newly collected webpages cover various categories (shopping, traveling, games and email). Please refer to the supplemental for the statistics of the dataset. We assign each webpage with one or more tasks selected from the 5 tasks, depending on the type of webpage. In particular, 71 webpages are assigned 1 task, 120 webpages are assigned 2 tasks and 9 webpages are assigned 3 tasks. To collect ground-truth saliency on the webpages under different tasks, we performed an eye-tracking experiment, following the experiment setup and methodologies in [24]. We recruited 24 participants for our experiment. In each viewing session, the participants are first informed of the task, followed by one or two webpages to perform the given task. For each webpage under each task, we collect eye-tracking data from 10 different participants, which are aggregated to produce the corresponding saliency

Table 2: Results for the ablation study. The best results are highlighted in red, while the second best are highlighted in blue.

Methods	KL ↓	sAUC ↑	NSS ↑
No task-specific subnet	1.330	0.576	0.412
No task-free subnet	0.810	0.628	0.559
Separate encoders	1.013	0.629	0.566
Separate CNNs	1.235	0.605	0.498
No pre-train on synthetic data	10.428	0.553	0.337
Train only on synthetic data	2.722	0.614	0.552
Ours	0.883	0.645	0.622

map. To the best of our knowledge, the newly collected dataset, containing 200 webpages, is the largest task-driven webpage saliency evaluation dataset (vs. 30 webpages in [24, 28]). Similar to previous works [3, 12, 14], we use the following metrics for evaluation: Kullback-Leibler divergence (KL), shuffled Area Under Curve (sAUC) and Normalized Scanpath Saliency (NSS).

5.2 Ablation Study

To evaluate the design of our network architecture and training strategy, we compare against the following baselines:

No task-specific subnet: We remove the task-specific subnet and concatenate the semantic vector of the input task label with the output of the shared encoder (before the task-free subnet) to predict task-driven saliency.

No task-free subnet: We convert our network to a one-branch architecture by removing the task-free subnet.

Separate encoders: Rather than using a shared encoder for the segmentation and task-free subnets, we use two separate encoders (VGG-16) for the two subnets.

Separate CNNs: We train 5 separate CNNs for each of the 5 tasks, and select the corresponding CNN for a given task, to predict the saliency.

No pre-train on synthetic data: We directly train our model on the real-world dataset, without pre-training the task-specific subnet on the synthetic data.

Train only on synthetic data: Instead of training on our real-world dataset, our model is only trained end-to-end on our synthetic data in Section 3.5.

Table 2 shows the results on our evaluation dataset. The results are obtained by averaging the metrics across all the tasks. (Please refer to the supplemental for the results on individual tasks.) Without the task-specific subnet, the performance is the worst. This shows that having a one-branch network to directly predict saliency from a webpage is not a promising solution and our task-decomposition framework is essential for the task-driven saliency prediction problem. The network without the task-free branch is slightly worse than our proposed network. This implies that while task-driven human attention mainly

Table 3: Performances of different saliency detection approaches on our evaluation dataset. The best results are in red, and the second best are in blue.

KL ↓

Method	Sign-	Form	Info.	Shopp-	Comm.	Average
	up	fill	brows.	ing	joining	
Human	0	0	0	0	0	0
Grad-CAM [25]	5.527	5.253	4.126	4.094	5.843	4.973
VIMGD [4]	2.513	2.726	2.987	5.462	3.127	3.363
SALICON [10]	0.651	1.116	0.569	0.771	0.595	0.739
SalNet [23]	1.129	1.893	1.041	0.941	1.028	1.207
Ours	0.867	1.152	0.731	0.861	0.812	0.883

sAUC ↑

Method	Sign-	Form	Info.	Shopp-	Comm.	Average
	up	fill	brows.	ing	joining	
Human	0.750	0.734	0.727	0.745	0.736	0.738
Grad-CAM [25]	0.519	0.533	0.503	0.507	0.512	0.515
VIMGD [4]	0.596	0.576	0.577	0.540	0.583	0.576
SALICON [10]	0.612	0.598	0.604	0.601	0.607	0.605
SalNet [23]	0.638	0.603	0.629	0.631	0.636	0.627
Ours	0.654	0.633	0.644	0.642	0.652	0.645

NSS ↑

Method	Sign-	Form	Info.	Shopp-	Comm.	Average
	up	fill	brows.	ing	joining	
Human	0.804	0.823	0.699	0.739	0.773	0.768
Grad-CAM [25]	0.144	0.214	0.008	0.085	0.112	0.126
VIMGD [4]	0.534	0.449	0.465	0.293	0.488	0.447
SALICON [10]	0.605	0.526	0.550	0.497	0.573	0.550
SalNet [23]	0.609	0.480	0.550	0.585	0.604	0.5652
Ours	0.646	0.594	0.624	0.607	0.638	0.622

focuses on the semantic components of webpages that are important to the task, it can still be attracted by other visual contents (*e.g.,* color and contrast) as in the free-viewing condition. Training task-specific models separately does not perform well, as compared with our unified model. With only the task-specific subnet (i.e., no task-free subnet), the model tends to put saliency mainly on task-relevant semantic components, but ignores the regions that people do look at (although with lower probabilities). This will result in a better KL score, which is more sensitive to the matching between high-saliency (probability) regions than between the low-saliency regions. In contrast, our full model learns to optimally allocate saliency between high-saliency task-relevant semantic components and other low-saliency regions. Therefore, although with a slightly worse KL score, it can better cover both high- and low-saliency regions, as reflected by other metrics. Finally, the results also suggest that our network can benefit from having a shared encoder for the segmentation and task-free subnets. This happens since the multi-task architecture can help our encoder learn better hidden representation to boost the performance of both tasks.

Without pre-training on the synthetic data, the performance of our model drops greatly. This confirms the importance of our task-driven data synthesis. In addition, learning with only synthetic saliency data does not perform well, due to the gap between the statistics of real and synthetic saliency data.

5.3 Comparison with Prior Works

We compare our method with several state-of-the-art works for free-viewing saliency detection, including one method for graphic design saliency, VIMGD [4], two recent methods for natural images, SalNet [23] and SALICON [10]. We also make comparison with a recent classification-driven concept localization model that is adapted to predict task-driven saliency by treating our task labels as class labels. For fair comparison, we finetune these models on the webpage saliency dataset [24] using the same training setting as ours. Unfortunately, we did not get the code for the free-viewing webpage saliency prediction method [28] for comparison. Thus, we make visual comparison with the results included in their paper (see the supplemental). For each webpage under each task, we run each method to get a saliency map. Since the free-viewing saliency detection methods do not take a task label as input, thus always producing the same results under different task conditions.

The results are shown in Table 3. Our model outperforms all the prior methods in sAUC and NSS, and achieves the second best performance in KL. The saliency detection models (SalNet, SALICON) generally perform better than other prior methods and SALICON even has a better performance than ours in KL. This is perhaps because that those free-viewing saliency models tend to fire at almost all the salient regions in a webpage, thereby generating a more uniform saliency distribution that is more likely to cover the ground truth salient regions. This leads to a higher KL score. However, such uniform saliency predictions certainly result in more false positives, making the performance of these models worse than ours in sAUC and NSS. The task-driven saliency method, Grad-CAM [25] performs worst in our evaluation dataset. This is likely because the complex and highly variable appearance of webpages make it difficult for classification-based models to find consistent patterns and identify discriminative features for different tasks, given our small dataset. Our model generally perform well in all metrics, which demonstrates the effectiveness of our model for predicting task-driven saliency. Human performance (Human) is also provided [12], which serves as upper bound performance.

Fig. 6 shows some qualitative results. Grad-CAM fails to locate salient regions for each task.The free viewing saliency models (*i.e.,* SalNet, SALICON, VIMGD) simply highlight all the salient regions, oblivious to task conditions. Hence, we only show one result from each of the prior methods regardless of the input task label. In contrast, given different tasks, our model can predict different saliency maps that are close to the ground truth. Please refer to the supplemental for more results.

6 Conclusion

We have presented a learning framework to predict webpage saliency under multiple tasks. Our framework disentangles the saliency prediction into a task-specific branch and a task-free branch. Such disentangling framework allows us to learn our model efficiently, even from a relatively small task-driven webpage saliency dataset. Our experiments show that, for the task-driven webpage saliency prediction problem, our method is superior to the baselines and prior works, achieving state-of-the-art performance on a newly collected dataset.

Acknowledgement: We thank the anonymous reviewers for their insightful comments. We also thank NVIDIA for donation of a Titan X Pascal GPU card.

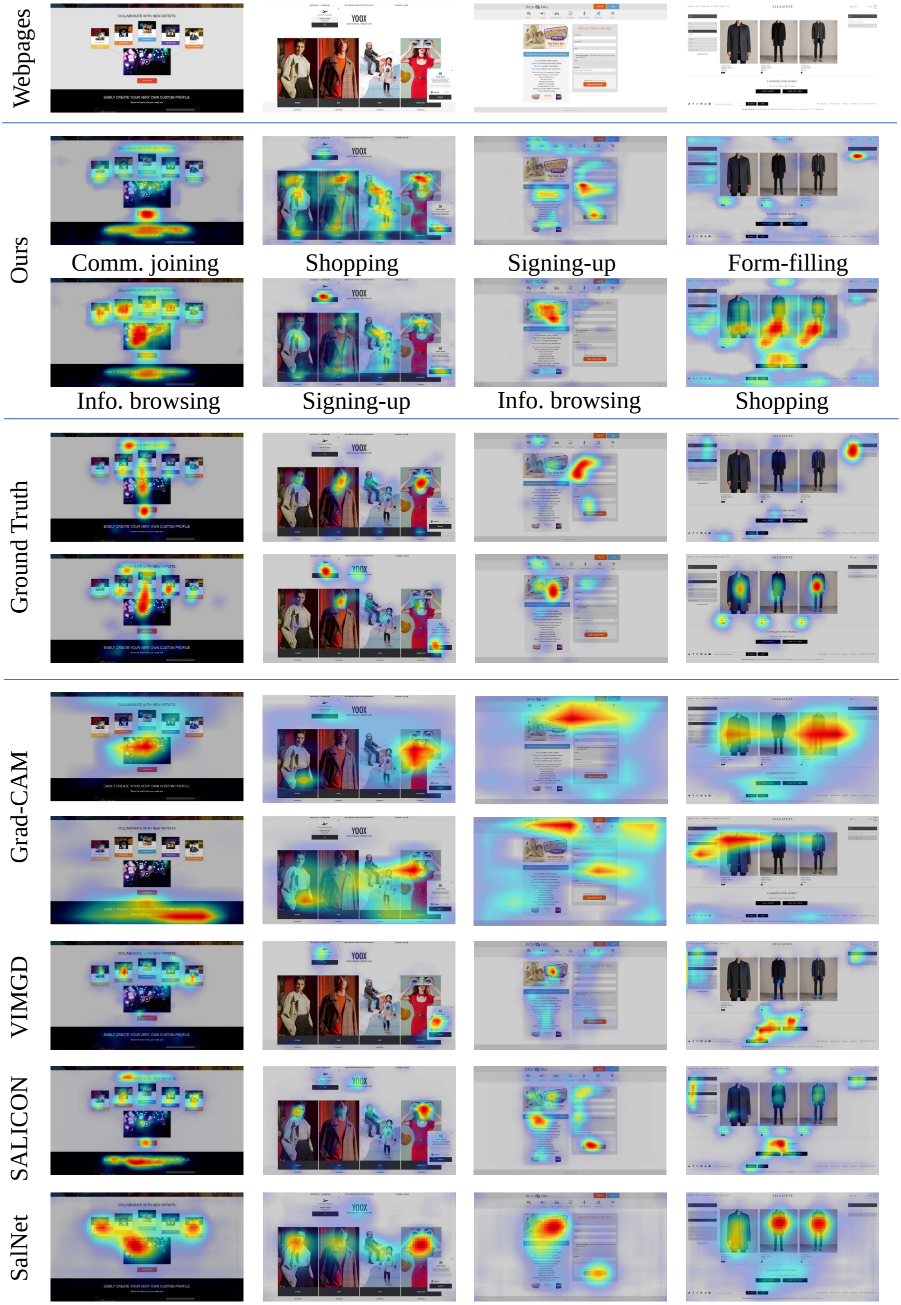


Fig. 6: Saliency prediction results of our method and prior methods under different task conditions.

References

1. Borji, A., Cheng, M., Jiang, H., Li, J.: Salient object detection: A survey. arXiv:1411.5878 (2014)
2. Borji, A., Itti, L.: State-of-the-art in visual attention modeling. TPAMI (2013)
3. Bylinskii, Z., Judd, T., Oliva, A., Torralba, A., Durand, F.: What do different evaluation metrics tell us about saliency models? arXiv:1604.03605 (2016)
4. Bylinskii, Z., Kim, N., O'Donovan, P., Alsheikh, S., Madan, S., Pfister, H., Durand, F., Russell, B., Hertzmann, A.: Learning visual importance for graphic designs and data visualizations. In: UIST (2017)
5. EYEQUANT: http://www.eyequant.com/, 2018
6. Guanbin Li, Yuan Xie, L., Yu, Y.: Instance-level salient object segmentation. In: CVPR (2017)
7. He, S., Jiao, J., Zhang, X., Han, G., Lau, R.: Delving into salient object subitizing and detection. In: ICCV (2017)
8. He, S., Lau, R.: Exemplar-driven top-down saliency detection via deep association. In: CVPR (2016)
9. Hou, Q., Cheng, M., Hu, X.W., Borji, A., Tu, Z., Torr, P.: Deeply supervised salient object detection with short connections. In: CVPR (2017)
10. Huang, X., Shen, C., Boix, X., Zhao, Q.: Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In: ICCV (2015)
11. Jiang, M., Huang, S., Duan, J., Zhao, Q.: Salicon: Saliency in context. In: CVPR (2015)
12. Judd, T., Durand, F., Torralba, A.: A benchmark of computational models of saliency to predict human fixations. In: MIT Technical Report (2012)
13. Judd, T., Ehinger, K., Durand, F., Torralba, A.: Learning to predict where humans look. In: ICCV. pp. 2106–2113. IEEE (2009)
14. Kruthiventi, S., Gudisa, V., Dholakiya, J., Venkatesh Babu, R.: Saliency unified: A deep architecture for simultaneous eye fixation prediction and salient object segmentation. In: CVPR (2016)
15. Kuen, J., Wang, Z., Wang, G.: Recurrent attentional networks for saliency detection. In: CVPR (2016)
16. Kümmerer, M., Theis, L., Bethge, M.: Deep gaze I: Boosting saliency prediction with feature maps trained on imagenet. In: ICLR (2015)
17. Lee, G., Tai, Y., Kim, J.: Deep saliency with encoded low level distance map and high level features. In: CVPR (2016)
18. Li, G., Yu, Y.: Visual saliency based on multiscale deep features. In: CVPR (2015)
19. Li, G., Yu, Y.: Deep contrast learning for salient object detection. In: CVPR (2016)
20. Liu, N., Han, J.: DHSNet: Deep hierarchical saliency network for salient object detection. In: CVPR (2016)
21. Navalpakkam, V., Itti, L.: Modeling the influence of task on attention. Vision research (2005)
22. N.Liu, J.Han: A deep spatial contextual long-term recurrent convolutional network for saliency detection. IEEE TIP (2018)
23. Pan, J., Sayrol, E., Giro-i Nieto, X., McGuinness, K., O'Connor, N.: Shallow and deep convolutional networks for saliency prediction. In: CVPR (2016)

24. Pang, X., Cao, Y., Lau, R., Chan, A.: Directing user attention via visual flow on web designs. In: SIGGRAPH Asia (2016)
25. Selvaraju, et al.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: ICCV (2017)
26. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. TPAMI (2017)
27. Shen, C., Huang, X., Zhao, Q.: Predicting eye fixations on webpage with an ensemble of early features and high-level representations from deep network. IEEE Trans. on Multimedia (2015)
28. Shen, C., Zhao, Q.: Webpage saliency. In: ECCV (2014)
29. Simonyan, et al.: Deep inside convolutional networks: visualising image classification models and saliency maps. In: ICLR Workshop (2014)
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014)
31. Tang, Y., Wu, X.: Saliency detection via combining region-level and pixel-level predictions with cnns. In: ECCV (2016)
32. Wang, L., Lu, H., Ruan, X., Yang, M.: Deep networks for saliency detection via local estimation and global search. In: CVPR (2015)
33. Wang, L., Wang, L., Lu, H., Zhang, P., Ruan, X.: Saliency detection with recurrent fully convolutional networks. In: ECCV (2016)
34. Xiao, H., Feng, J., Wei, Y., Zhang, M., Yan, S.: Deep salient object detection with dense connections and distraction diagnosis. IEEE Transactions on Multimedia (2018)
35. Xu, Y., Wu, J., Li, N., Gao, S., Yu, J.: Personalized saliency and its prediction. In: IJCAI (2017)
36. Yarbus, A.: Eye movements during perception of complex objects. In: Eye movements and vision (1967)
37. Zhang, et al.: Top-down neural attention by excitation backprop. In: ECCV (2016)
38. Zhang, P., Wang, D., Lu, H., Wang, H., Yin, B.: Learning uncertain convolutional features for accurate saliency detection. In: ICCV (2017)
39. Zhang, X., Wei, Y., Feng, J., Yang, Y., Huang, T.: Adversarial complementary learning for weakly supervised object localization. In: CVPR (2018)
40. Zhao, R., Ouyang, W., Li, H., Wang, X.: Saliency detection by multi-context deep learning. In: CVPR (2015)
41. Zhou, et al.: Learning deep features for discriminative localization. In: CVPR (2016)