UEyes: Understanding Visual Saliency across User Interface Types

ANONYMOUS AUTHOR(S)

Graphical user interfaces display elements such as images and text in a grid-based layout. However, different UI types exhibit significant differences in the number of elements and how they are displayed. For example, webpages make heavy use of images and text, whereas desktop UIs tend to have a lot of small images in comparison. How do such differences affect the way users look at UIs? To understand this question, we collected and analyzed *UEyes*, a large eye-tracking dataset (62 participants; 1980 UI screenshots) covering four basic UI types: webpage, desktop, mobile, and poster. We analyze differences in common bias such as color, location, and gaze direction. We also compare state-of-the-art predictive models and propose improvements that better capture typical tendencies across UI types. Our dataset and models are publicly available.

CCS Concepts: • **Human-centered computing** → **Empirical studies in ubiquitous and mobile computing**; • **Computing methodologies** → *Computer vision.*

Additional Key Words and Phrases: Human Perception and Cognition; Interaction Design; Computer Vision; Deep Learning

ACM Reference Format:
Anonymous Author(s). 2023. UEyes: Understanding Visual Saliency across User Interface Types. 1, 1 (December 2023), 28 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Understanding what grabs the user's attention when looking at user interfaces (UIs) is a long-standing research topic in HCI. This is essential for designers to guide users' attention, convey critical information, and avoid visual clutter [75, 81]. However, despite several years of work on this topic, we have only a minimal view of how different *types* of UIs differ in visual saliency. For instance, posters often consist of a few graphical images, while desktop and mobile UIs often have more components structured as widgets. Therefore, understanding how such differences carry over to eye movement patterns is essential. The hypothesis underlying this work is that the visual features of the UI should be reflected in the users' gaze patterns.

This paper takes a two-pronged approach to advance the understanding of eye movements related to different UI types. First, we collect and analyze *UEyes*, a novel eye-tracking dataset captured by a high-fidelity in-lab eye tracker on a large scale. While previous work has used mouse movements or manual annotations as a proxy for eye movements, UEyes offers access to elaborate ground-truth data on visual saliency. Our dataset offers multi-duration saliency maps and scanpaths of 62 users who looked at 1980 different UIs, including 495 desktop UIs, 495 mobile UIs, 495 webpages, and 495 posters. In this paper, we analyze and compare saliency-related tendencies across the UI types, including bottom-up factors based on the visual primitives of the stimulus, such as color bias, top-down (learned) factors based on

The word count of this paper is 9019.

the distribution of features in the dataset, such as location bias and scanpath direction. We present several previously unreported findings on what distinguishes the different UI types.

Second, we use the dataset to assess and improve computational models of visual saliency. Given a UI as input, a saliency model can predict saliency maps or scanpaths simulating how users perceive the UI. These models are helpful for UI designers because they can predict where users are likely to fixate within a given UI design and update it better to emphasize the essential areas of their UI design. Such models may help 'reflow' UI designs and create versions for different screen sizes that maintain the desired visual emphases.

High-quality datasets are needed for data-driven approaches to employ modern computational models effectively (*e.g.,* based on Deep Learning) and improve our understanding of visual saliency. There is a plethora of work on saliency modeling, predicting where viewers look at [21, 36, 39, 41, 48, 51, 55, 68] and numerous scanpath models, predicting gaze over time [2, 3, 33, 47, 64, 89]. A limitation of current approaches is that they only work well when domain-specific data is available. Moreover, existing datasets are relatively small (*e.g.,* MASSVIS [11] and iSUN [96]) and often limited to specialized types of designs (*e.g.,* only mobile UIs [53]). Our UEyes dataset includes high-quality eye-tracking data for various user interface (UI) types, including webpages, mobile UIs, desktop UIs, and posters, making it more generalizable and valuable for a broader range of applications. In addition, although Leiva et al. [53] proposed analysis for mobile UIs, to our knowledge, no prior work has analyzed biases in saliency maps (*e.g.,* location bias) and in scanpaths (*e.g.,* saccade angle) to compare different UI types. We aim to fill this gap by systematically analyzing and comparing eye-tracking data across different UI types.

Furthermore, our UEyes dataset enables dedicated models to predict visual saliency and scanpaths across different UI types. A multi-type dataset is important because accuracy decreases significantly when tested on UI types not contained in the training data. Designers could use these models to improve the user experience of their interfaces. With visual saliency models, designers can improve their designs by making informed decisions about how users are likely to view their UIs [15]. Scanpath predictive models, unlike saliency maps, retain information about the order of fixations and their temporal dynamics. This information is important to retain in applications. For example, Scanpath predictive models allow designers to understand visual flows and adjust their designs to encourage users to view the UIs in the correct order [69].

The most relevant prior research to ours is UMSI [29], which proposed a crowdsourced dataset (Imp1k) and a unified model of saliency map and importance trained on images from different design classes, including webpages, movie posters, mobile UIs, infographics, and advertisements. UMSI created a generalizable visual importance model and performed well among various design types. However, UMSI did not further report differences in how users view the different types. To accomplish that, we collect and classify images based on common UI types and further introduce a systematic eye-tracking analysis and comparison across UI types. Unlike UMSI, we collect real-time eye-tracking data via an eye-tracker. Although crowdsourcing approaches enable the collection of large eye-tracking datasets (*e.g.,* Imp1k and SALICON [39]) with proxies of eye-tracking data, such as cursor-based or webcam-based methods, they cannot simulate the same results as collected by actual eye trackers. Webcam-based approaches suffer from low accuracy, while cursor-based methods reflect different cognitive processes from eye movements [83].

In sum, this paper makes three notable contributions:

(1) We present the first analysis and comparison of eye movements across common UI types. We report differences regarding location bias, color bias, saccade angle and amplitude, and visited vs. revisited elements.

(2) We compare existing predictive models for saliency maps and scanpaths on how well they perform across the UI types. We make improvements to existing models informed by our data, such as changes in loss terms, training strategies, and modeling features (*e.g.,* inhibition of return).
(3) We release the largest in-lab eye-tracking dataset (62 participants, 1980 UI screenshots) with associated metadata and eye-tracking logs categorized in webpages, desktop UIs, mobile UIs, and posters.

2 RELATED WORK

Predicting where people look at is paradigmatically more ambiguous than typical Computer Vision tasks such as image segmentation [63] or object detection [42], among other areas. We hypothesize that we should observe significant differences among UI types for the same reasons that considerable differences have been reported for different scenes and individuals. Differences among individuals and stimulus types can be attributed to the physiologically determined bottom-up factors and the learned top-down features [100]. On the one hand, the biological basis for bottom-up saliency is rooted in the parallel processing of retinal input in the visual cortex [85]. Bottom-up features are constituted by a few physiologically determined visual primitives – size, color, shape, orientation, and motion [54, 92]. For example, contextually unique objects along these features tend to draw attention. Larger objects, which also have more stimulus energy, also have higher saliency. On the other hand, top-down factors include task goals and expectations. Expectations are formed over repeated exposure to instances of a type of stimulus [78].

2.1 Visual Saliency in Natural Scenes

Previous work on visual saliency outside HCI has focused on non-UI stimuli and natural scenes. Consequently, viewing patterns reported for them may not hold for UIs. The research literature looking at the saliency of natural scenes has found several replicated effects, or (viewing) biases, which we revisit in this paper:

Center bias: Studies have reported a bias toward looking at the center of the screen when viewing natural scenes [35, 65]. The effect has been replicated with artificial media, especially video [59], text [73], and single objects [65]. Whether this holds for UIs is unclear since much of their most informative elements lie in the upper half of the display.

Horizontal bias: In looking at natural images featuring objects, fixations tend to be distributed more horizontally than vertically [65, 66]. Again, UIs differ from natural scenes in that they mostly organize the information vertically, not horizontally. Therefore, we might see this effect weaken.

Color bias: Color brightness and contrast have been mentioned among the primary features driving bottom-up saliency [27, 32]. Visual designs such as websites and mobile UIs typically contain colorful icons and images perceived as highly salient. Thus, we would expect this effect to hold.

2.2 Visual Saliency in UI Designs

Research on visual saliency in HCI has looked at either eye-movement data but is limited to a single UI type (e.g., mobile UIs [53]) or proxy constructs correlated with eye movements but not desired for saliency modeling. *Visual impression* refers to the reported visual appeal of graphical regions or objects on a UI measured via rating scales, with results reported for both desktop [56] and mobile interfaces [61]. However, visual saliency is a construct related to the control of visual attention, not self-reports of what is felt to be important.

A concept closely related to saliency is that of *visual importance*. Bylinskii et al. [15] extended a pre-trained neural network [79] for predicting which regions in a graphic design are felt to be more critical. Importance was measured utilizing cursor exploration of a blurred page. However, a "poor man's eye tracker" [19], which involves an element of the reflective judgment of importance, is not a good proxy for gauging visual saliency [83]. Finally, research on *visual clutter* is directly motivated by theories of saliency. The work of Rosenholtz [75] showed how models of visual saliency could be exploited to compute indices of how cluttered a display is perceived.

2.3 Visual Saliency Datasets

Many existing visual saliency datasets are either limited to specialized types of designs and a relatively small number of saliency results. Most of them only contain one specific type of visual design collected from a particular set of participants, such as visualization (e.g., MASSVIS [11]), indoor and outdoor natural images (e.g., iSUN [96], SALICON [39], MIT1003 [43], MIT300 [41], NUSEF [72]), mobile user interfaces [53]), visual flows on comics [16], webpages [80], or posters [67]. CAT2000 [9] contains 20 categories, but these categories are different classes of natural images, with additional augmented natural images including the non-photorealistic rendering of natural images such as sketches and cartoons, and noisy natural images, such as low-resolution scenes and Gaussian-noised images. UEyes, the dataset we have collected for this work, contains eye-tracking data on four common categories of UIs, and a wide variety of images focusing on visual designs.

Prior work explored the crowdsourced collection of saliency-related data, e.g., Imp1k [29] and SALICON [39]. Crowdsourcing, however, excludes the use of high-fidelity in-lab eye trackers. It has hence used proxy sensors, such as cursor movements or webcams. Webcam-based approaches [96] often suffer from lower accuracy since errors may occur during facial landmark tracking, eye region extraction, and calibration with the webcam. Cursor-based approaches [4, 39, 44, 45] are slower, more deliberative cognitive processes than eye movements.

2.4 Computational Visual Saliency Models

Given a stimulus image, a computational model of visual attention predicts a saliency map [7] or a scanpath showing the order in which eye fixations would take place over the image [53]. Stimulus-driven saliency models are computed using visual primitives [8, 10]: They work well for first-time views a user has not been exposed to [31, 37]. In contrast, task-driven models gauge a user's familiarity [78] being affected by expectations, location memory, and search strategies. Data-driven models make predictions based on image features and have architectural assumptions that allow them to capture domain-specific viewing tendencies [53] better.

Computational modeling of saliency has been a topic of interest in computer vision and HCI since the work of Itti and Koch [37]. Recent research on saliency maps has explored emerging types of deep learning architectures. An early approach was an ensemble of deep networks (eDN) [87], using deep nets as extractors for hierarchical features and combining the outputs with a support vector machine. DeepGaze I [48] followed the same logic, considering a sparsification loss term, center bias, and a smoothing kernel. ML-Net [20] fine-tuned the features on saliency prediction to improve the previous two models. Saliency Attentive Model (SAM) [21] added temporal tuning by employing progressive formation of saliency with ConvLSTM blocks to process features.

To consider the evolution of saliency maps over time, Fosco et al. [30] proposed a multi-duration saliency, *i.e.,* predicting saliency for different durations. Generative adversarial networks (GANs) also obtained a good approximation of saliency distributions [17, 68]. Some models improved the predictive performance by exploiting contextual information and encoding the similarity between images [46, 58, 71]. SalFBNet [25] employed a recursive feedback architecture

feeding latter computation blocks into the earlier stage of the computations, being useful in recognition tasks compared to purely feedforward networks [97]. Nonetheless, similar results could be achieved by increasing the capacity of networks. For example, EML-NET [38] performed a multi-branch prediction at the decoding stage. UniSal [26] unified the prediction of saliency on both image and video stimuli. DeepGazeIIE [55] employed a combination of multiple backbones.

Scanpath prediction is a more challenging problem since information on the order of fixations must be retained. Itti and Koch [37] implemented an inhibition of return (IOR) mechanism to generate a sequence of fixations using the computed saliency maps. This work inspired a group of techniques that utilize a saliency map for scanpath generation. For example, Tavakoli et al. [74] proposed a joint sampling mechanism to estimate the saliency and gaze points. Wloka et al. [91] improved Itti's saccade generation scheme by considering the high-level saliency estimated with deep nets and a peripheral conspicuity map obtained using low-level saliency approaches. Chen and Sun [18] introduced an advanced architecture to learn the inhibition of return maps from data. Xia et al. [94] jointly estimated the saliency and fixation location with an auto-encoder in a framework mimicking [74].

Other recent methods have worked on scanpath models that can generate a sequence of fixation locations. For example, Verma and Sen [86] employed a recurrent architecture to generate a sequence of fixations in a grid-based representation. PathGAN [3] used GAN-based training to estimate a fixation sequence with location and duration. In this work, we compare a few well-known predictive models for saliency maps and scanpaths in their capability to model observed differences among UI types.

3 DATASET: UEYES

UEyes is a dataset consisting of 1980 UI screenshots and associated metadata and eye-tracking logs from 62 viewers collected in a laboratory using a modern eye tracker. Altogether, the dataset contains 495 screenshots from each UI type:

Webpage: We collected 494 webpage images from the Alexa 500 dataset [90], 1507 images from the Visual Complexity and Aesthetics dataset [62], and 200 images from the imp1k dataset [29]. We further extended the webpage image set by taking 103 additional webpage screenshots.

Desktop UI: The desktop UI image set contains the Waltteri Github desktop UI dataset [23] with 51 desktop UIs and our collected additional 303 desktop UI images.

Mobile UI: We sampled 1761 images out of 46064 mobile UI images from the RICO dataset [24]. We extended it with 42 of our collected mobile UI images.

Poster: The poster image set contains 200 ads and 198 infographics from the imp1k dataset [29] with additionally collected 103 posters.

The additional images we collected besides the ones in existing datasets are either substantially different from the existing ones or are widely used in daily life, such as music and university apps. This was to ensure a diverse and representative dataset. Additionally, some desktop images were added to ensure the final dataset was balanced. Images containing pornography were filtered out, and then the images of each type were pooled together and sampled randomly to create 'image blocks' for user assessment so that each block includes 9 images from each UI type and 36 images in total. We created 55 blocks for our study.

During the data collection process, the screen angle was adjusted for each participant to mimic their typical viewing experience. Participants sat approximately 50–65 cm from the screen, and the same visual angle was used for all UI types, including mobile UIs, to ensure a fair comparison. This allowed for consistent data collection and analysis across

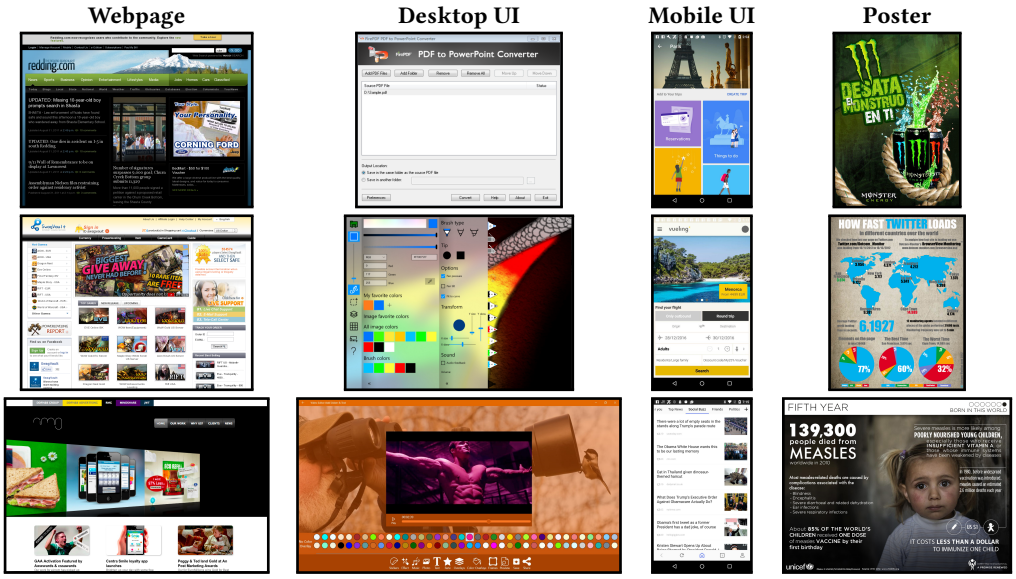


Fig. 1. Examples of user interfaces in the UEyes dataset. The full dataset contains 495 images of each UI type: webpage, desktop UI, mobile UI, and poster.

different UI types. Presenting all UI types in a consistent manner ensures that the limits of tracking accuracy does not disproportionately affect the mobile UI type.

3.1 Participants

We recruited sixty-six participants (23 male, 43 female) via mailing lists and promotions on social media. The average age was 27.25 (SD = 7.26). Participants had normal vision (43) or corrected-to-normal vision: wearing glasses (18) or contact lenses (5). No participant was color blind. We dropped 4 users' gaze data due to inaccurate eye-tracking calibration. The study took one hour for each user, who received a compensation of 30 EUR.

3.2 Experimental Design

Our system randomly sampled 9 blocks of 36 images for each user (out of the pool of 55 blocks, see above). One block includes 9 images from each UI type. Images in each block were presented in a randomized order.

3.3 Apparatus

The images were shown on a desktop monitor (HP Compaq LA2405wg, 24 inches). The monitor has a dimension 32.5 × 52 cm with a resolution of 1920 × 1200 px. We used a Gazepoint GP3 eye tracker with a sampling rate of 60 Hz to collect high-quality gaze data. The eye tracker was placed under the screen and tilted upwards. The angle was adjusted to suit the individual participant. The participants sat approximately 50–65 cm from the tracker, ensuring that the eye-tracking software (Gazepoint Control) indicated a desirable distance.

3.4 Procedure

The eye tracker was first calibrated with Gazepoint Control's 9-point calibration and tested on the calibration test screen. After calibration, the participant was shown three images of different-sized grids and was instructed to look at the corners of the grids, starting from the top-left corner and moving clockwise. (This was done for quality control at the post-processing stage.) After calibration, participants completed nine blocks (see above) with self-managed breaks. The participant looked at each presented UI image for 7 seconds and was asked to examine the images like they would in a real-world situation. Like in other bottom-up saliency studies, no specific task was given. After the last block of UI images, the participant filled out a demographics questionnaire.

3.5 Data Processing

We double-checked the collected data to guarantee the dataset's quality and removed the user data with inaccurate calibration or duplicated results. The final dataset contains 94.86% of the originally collected data. Fixations outside the images (6.8% of the fixations) were not considered for analysis. We include the details of the UEyes dataset in *Supplementary Materials*.

4 FINDINGS

In the following, we analyze the data regarding location bias, color bias, saccade angle and amplitude, and visited vs. revisited elements across all UI types.

4.1 Effect of Location

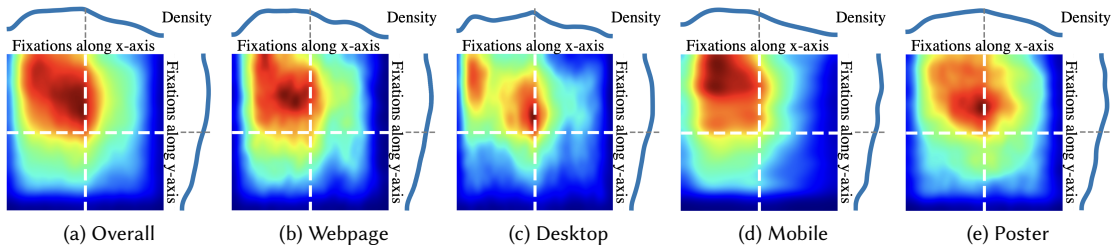


Fig. 2. Location bias: distribution of fixations along normalized screens. Compared to the center bias of natural images, fixations on user interfaces are mostly located in the top-left area.

Figure 2 shows location bias for different UI types and Figure 3 displays the corresponding quadrant distribution of fixations. The location bias was computed by normalizing the saliency distribution relative to the individual UI image size and then aggregating all the UI saliency results belonging to each UI type. Overall, compared to the known center bias of natural images [13], we noticed that the top-left quadrant tends to attract more fixations than the other quadrants across all UI types, indicating that participants pay more attention to the top-left area on UIs. Fixations on webpages, mobile UIs, and posters are spread in the entire top-left area, while for desktop UIs, the salient regions are separated into two parts: One right above the center of desktop UIs and the other around the top-left corner. The most salient area of webpages is around the center-right of the top-left quadrant, and for mobile UIs, the top part of the

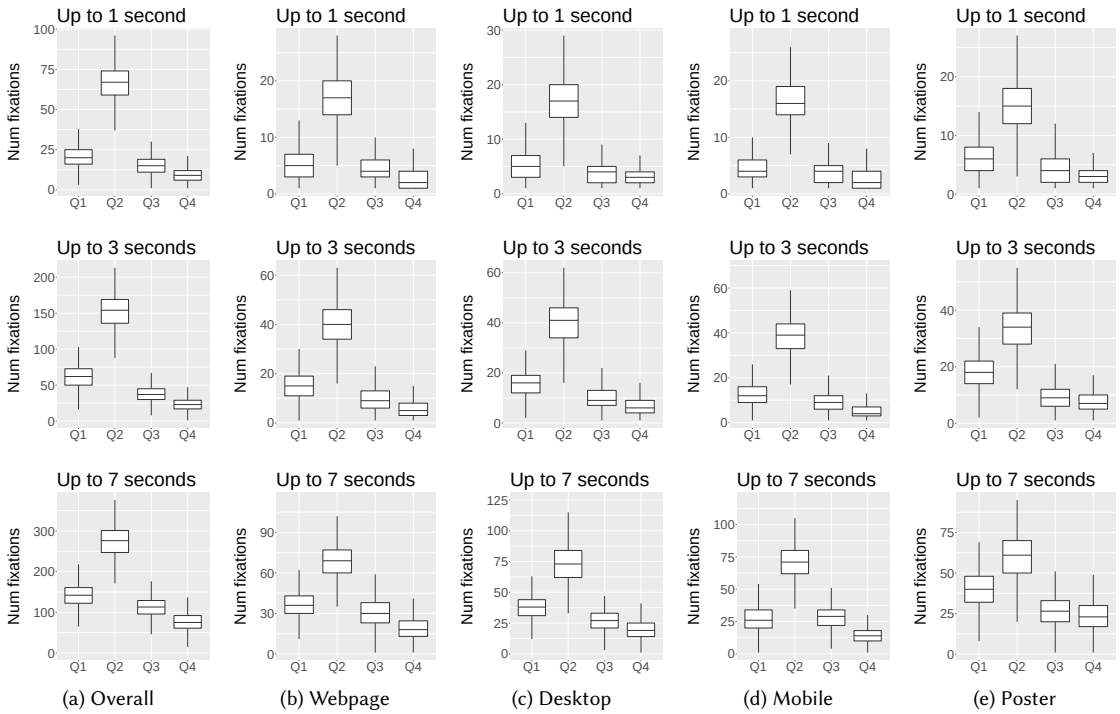


Fig. 3. Location bias: quadrant distribution of fixations. The top-left quadrant tends to attract more fixations than other quadrants across all UI types.

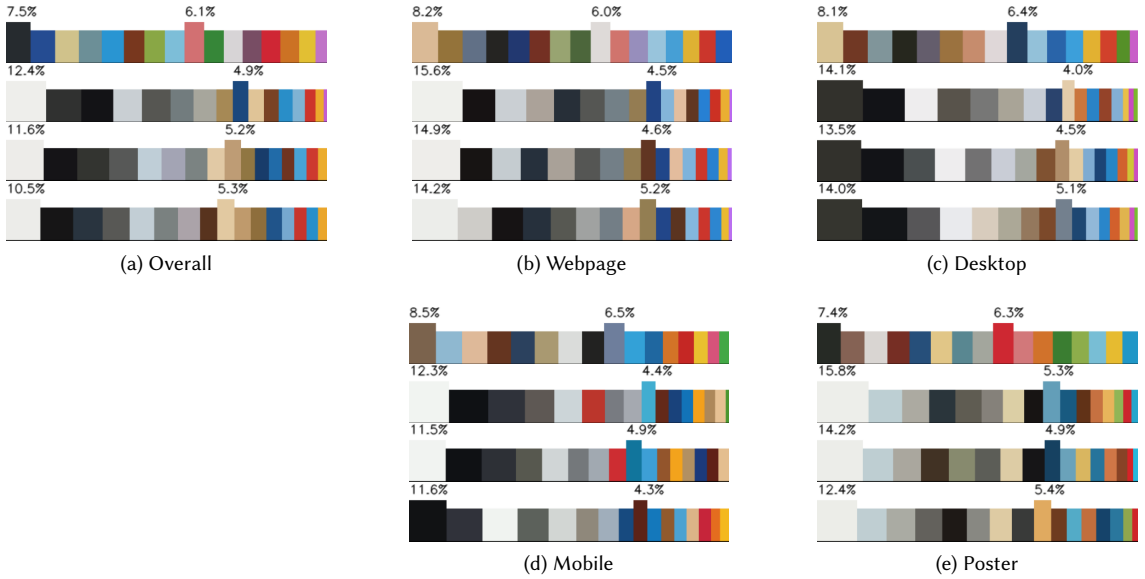
top-left quadrant attracts the most attention. Differently for desktop UIs and posters, the most salient area appear right above the center of the UIs.

An omnibus test revealed statistically significant differences between the average number of fixations per user and visual content for each quadrant (Q1: top-right, Q2: top-left, Q3: bottom-left, Q4: bottom-right). For example, in the overall condition: $\chi^2(3) = 183.930, p < .0001$. Similar results were obtained for each type of UI.

We then ran Bonferroni-Holm corrected pairwise comparisons as post-hoc tests, and found that the difference between Q1 vs. Q2 was statistically significant in all cases ($p < .001$). The difference between Q1 vs. Q3 and Q1 vs. Q4 was statistically significant when users viewed the images for 3 or more seconds ($p < .001$), The difference between Q2 vs. Q3 and Q2 vs. Q4 was statistically significant in all cases ($p < .001$). Finally, the difference between Q3 vs. Q4 was significant when users viewed the images for 3 or more seconds ($p = .018$).

4.2 Effect of Color

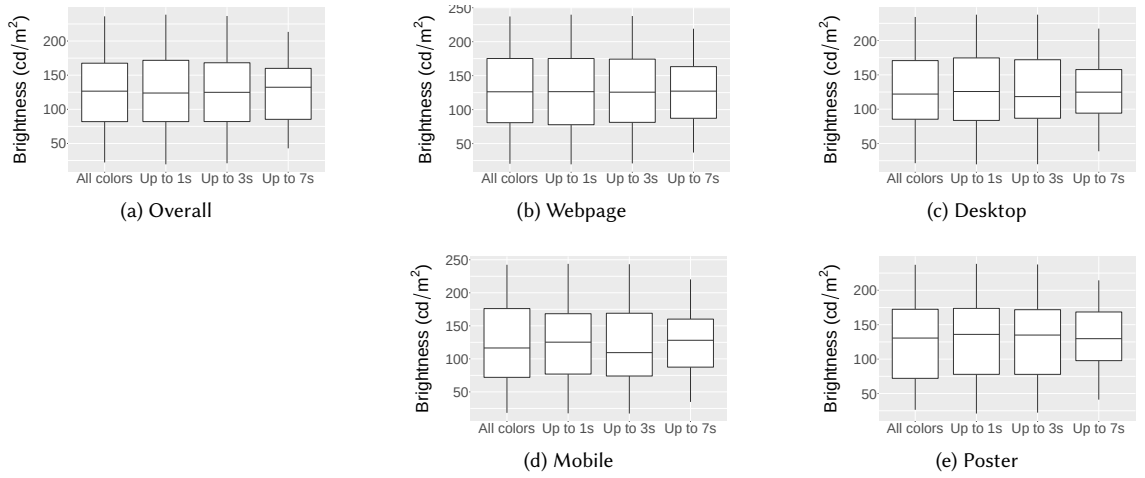
We show color bias across different UI types in Figure 4. The top color bar shows the 16 most prevalent colors in the original UI images for different UI types. The other color bars rank the top 16 colors by the number of fixations on those colors, sorted by frequency. We computed the 16 most prevalent colors using k-means clustering, therefore similar colors are merged together. Figure 5 shows a comparison between the displayed colors (shown as "all colors" in the plots) and the fixated colors. On average, brighter colors attract more attention than darker ones by comparing the



(a) Overall (b) Webpage (c) Desktop

(d) Mobile (e) Poster

Fig. 4. Color bias: the 16 most prevalent colors on UIs (top row), and the top 16 fixated colors sorted by frequency up to 1 s of fixations (second row), up to 3 s (third row), and up to 7 s (bottom row).



(a) Overall (b) Webpage (c) Desktop

(d) Mobile (e) Poster

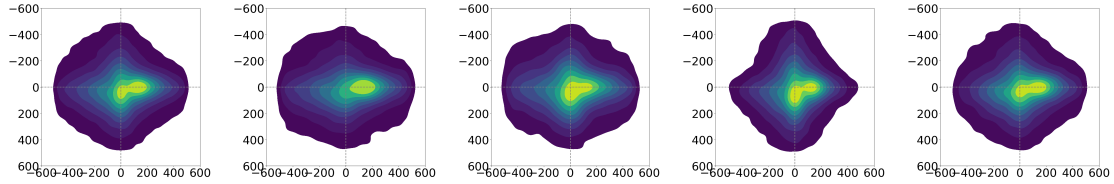
Fig. 5. Color brightness bias compares the brightness of all the displayed and fixated colors. Overall, brighter colors tend to attract slightly more attention than darker ones, especially for short time spans.

displayed and fixated color brightness. Webpages, desktop UIs, and mobile UIs seem to draw more attention to brighter color areas relative to the displayed colors. The only exception is posters, where the average brightness in fixated colors is lower than in displayed colors. However, the most frequent color on the posters where participants look is still a light

color. Although desktop UIs have higher average brightness of fixated colors than displayed colors, the top three colors attracting the most fixation points are dark colors.

To further investigate whether a reliable effect exists, we computed the pixel brightness values by using sRGB Luma coefficients (ITU Rec. 709) [6], which reflect the corresponding standard chromaticities, and compared the distribution of fixation and non-fixation brightness values. Bartlett's test of homogeneity of variances was not statistically significant, neither for all UIs combined ($\chi^2(3) = 1.003, p = .8004$) or for each UI type individually ($\chi^2(3) \leq 0.832, p \geq .8416$). Therefore we conclude that color does not significantly affect visual saliency.

4.3 Saccade Angle and Amplitude



(a) Overall (b) Webpage (c) Desktop (d) Mobile (e) Poster

Fig. 6. Saccade bias analysis shows the direction and distance between consecutive fixation points. Gaze directions are mainly directed at the right or bottom part of the UIs, with larger distances towards the right part. Users prefer to look from left to right (with larger gaze movement) and from top to bottom.

Saccade angle and amplitude reveal the tendency and speed of eye movements, which can be utilized to optimize the placement of UI elements and the flow of information in a UI. By understanding these metrics, designers can arrange their designs in a way that aligns with the natural gaze behavior of users, potentially leading to a better user experience. Figure 6 shows the distribution of direction and distance between two consecutive fixation points, represented by the saccade angle and amplitude in polar coordinates. We can see that, overall, gaze directions are mainly oriented towards the right or bottom part of the UIs. However, UI types differ markedly in this respect. Users prefer to look more from left to right on webpages than other UI types. Similarly, users tend to scan from left to right on posters, with a small number of directions towards the bottom, but the distances of gaze moved to the right were more diverse. In contrast, users look both from left to right and top to bottom on desktop UIs and mobile UIs. The distances towards the right part are larger than distances to the bottom on desktop UIs. On mobile UIs, they remain about in the same range.

A Kruskal-Wallis Chi-squared test was statistically significant for all UI types (e.g. $\chi^2(3) = 484.41$, $p < .0001$ for the overall category) so we ran pairwise comparisons (Bonferroni-Holm corrected) as post-hoc tests and found that all directions were significantly different from each other for all UI types; the right direction is the most frequent, followed by the left, bottom, and top directions.

4.4 Visited vs. Revisited Elements

We segmented the UIs and classified the UI elements into three categories: image, text, and face, by extending the functionality of the UIED model [95]. We then counted the number of elements in each category that were visited (fixated) and revisited (fixated again). An element is considered revisited if it differs from the previously fixated element and has received at least three fixation points. The results are shown in Figure 7.

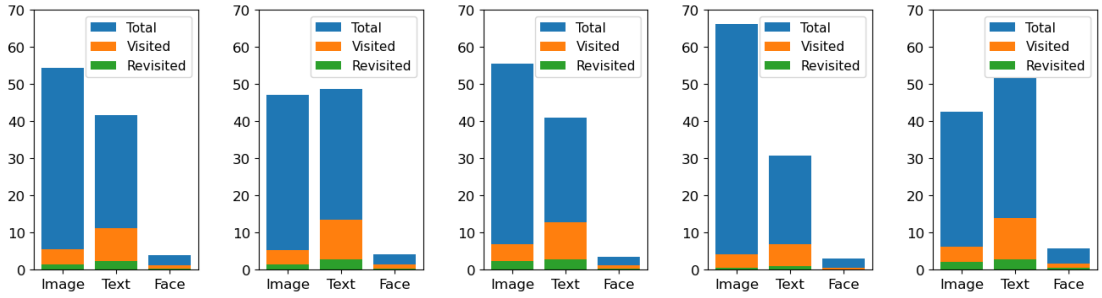


Fig. 7. Visit vs. revisit bias analysis showing the ratios of visited and revisited elements in different element categories. Text elements are more likely to be fixated and re-fixated than images

We observed that text elements have a higher probability of being fixated than images. While desktop UIs had many small images (like icons) compared to text, we noticed the opposite in posters, i.e., they had a few (typically one or two) large images. Webpages had about the same number of images and text. It is worth noting that mobile UIs have lower visiting and revisiting ratios than other UI types, indicating that participants have less chance to go back to look at the same content.

Differences in visit and revisit ratios were found to be statistically significant between the three element types (image, text, face) for all UI types. For example, a comparison between the visit ratios for the overall condition: $\chi^2(2) = 9.295$, $p < .01$. Post-hoc pairwise comparisons (Bonferroni-Holm corrected) revealed statistical significance for all the UI types compared. We conclude that text attracts fixations the most, followed by images and faces.

4.5 Summary

The top-left quadrant tends to attract most fixations and brighter colors do not attract significantly more fixations than less bright colors. When looking at UIs, saccade directions mostly move from left to right and top to bottom. Participants tend to spend more time looking at text elements than images, which explains why the saccade directions tend to move from left to right. Overall, our findings on mobile UIs align with the results of Leiva et al. [13]. We introduced more analysis metrics and found more biases for all the UIs types. We report the following characteristics and differences per UI type:

Webpage: Participants prefer to scan more from left to right on webpages with larger distances between consecutive fixations than other UI types.

Desktop UI: Instead of having fixations spread in the top-left quadrant, the salient areas of desktop UIs are separated into two parts: right above the center and around the top-left corner.

Mobile UI: Mobile UIs have lower visiting and revisiting ratios than other UI types. It indicates that participants tend to focus more on the few most attractive UI elements, ignore others, and have less chance to go back to look at the same elements.

Poster: Compared to desktop UIs and mobile UIs, participants have much more intention to scan from left to right, with only a small portion of saccade directions from top to bottom. The distances of the consecutive fixation points have a more significant variation than other UI types.

5 ASSESSING SALIENCY MAP MODELS

We compare data-driven saliency map predictive models in light of the reported differences in UI types. We consider the state-of-the-art traditional optimization-based model (GBVS) and data-driven models (SAM, UMSI) along with UMSI++, an improved version we propose:

Graph-Based Visual Saliency (GBVS) [34]. is a bottom-up saliency map model detecting informative features depending on the entire image. It takes the saliency-based visual attention model proposed by Itti and Koch [37] to extract visual features computed by linear center-surround operations with Gaussian pyramids for intensity, color, and orientation. It then forms graph-based activation maps on visual features and normalizes them to highlight conspicuity. The global visual feature extraction and graph-based activation maps enable the model to capture saliency maps at the global level, which is more efficient than prior approaches relying on local information.

Saliency Attentive Model (SAM) [21, 22]. incorporates an attentive convolutional long-short term memory (Attentive ConvLSTM) saliency map model to focus on different spatial location features to enhance predictions sequentially. The model iteratively and progressively refines the predicted saliency map results via the LSTM architecture. SAM learns a set of prior maps generated with Gaussian functions to learn saliency priors, such as the center bias typical of human eye fixations, without needing hand-crafted prior information to obtain improved feature extraction capabilities.

UMSI [29]. is a unified model of saliency map and importance trained on images from different design classes, including posters, infographics, mobile UIs, and natural images. It uses an encoder-decoder architecture and aggregates image information at multiple scales to predict visual importance in input graphic designs. It employs an automatic classification module for the input graphic designs to better capture the saliency patterns with class-specific information. UMSI was trained on the dataset of visual importance from cursor-based crowdsourcing data. The cursor is a good proxy for eye-tracking but still cannot simulate the same results as data collected by eye trackers.

UMSI++ and SAM++ (Ours). is a variant we created by employing new loss terms and a two-step training process. The main module of the original UMSI model was trained with KL-divergence [40] and Cross-Correlation [52] losses with coefficients 10 and -3. The output of the UMSI model is the flipped saliency maps requiring postprocessing of black-to-white invert. Our UMSI++ model employs an end-to-end joint training process by refining the model via different loss terms. During the first 10 epochs of training, the model approaches the ground-truth saliency maps using the Mean Squared Error (MSE) loss between the predicted and the ground-truth saliency maps. This helps the model to accurately predict the saliency maps. For the remaining epochs, the model is trained using a combination of loss terms, including the KL-divergence and Cross-Correlation loss terms [52] used in UMSI, as well as two additional loss terms: the Normalized Scanpath Saliency (NSS) loss and the Similarity loss. The NSS loss evaluates the average normalized saliency at fixation points, while the Similarity loss measures the intersection between the predicted and the ground-truth saliency maps. These loss terms help the model better capture fixations and improve its overall performance. KL-divergence and Cross-Correlation are distribution-based because they focus on the continuous distributions of the saliency maps, rather than on individual points or locations. In contrast, NSS and Similarity are location-based because they focus on the locations of fixation points in the saliency maps. Togehter, these loss terms have been shown to perform well in predicting fixation points, and can help the model better capture eye fixations [12, 99]. Computation details are given in the *Supplementary Materials*. The training takes about 1 hour on one NVIDIA GeForce RTX 2080Ti

GPU. For comparison, we apply the same training pipeline and loss terms to the SAM architecture to get the result of the model SAM++.

5.1 Evaluation Metrics

To evaluate the accuracy, we used six widely used evaluation metrics.

Area under ROC Curve (AUC). is the most commonly used evaluation metric for measuring saliency map performance. It evaluates the saliency map as a binary classifier of fixation points at various thresholds. Receiver Operating Characteristic Curve (ROC Curve) is a curve showing the rates of the actual positive points and the false positive ones at different discrimination threshold values. AUC is defined as the area under such a curve measuring the true and false positive rates under the binary classifier, which can be computed by taking the integral of the area under the ROC curve in practice. AUC-Judd [14, 43] is a variation of AUC. The true positive rate is defined as the ratio of the true positive points to the number of ground-truth fixation points above various threshold values. The false positive rate is the ratio of false positive points to the total number of non-fixation pixels.

Normalized Scanpath Saliency (NSS) [70]. is the average normalized saliency at fixation points. NSS is more sensitive to detecting false positive points than the AUC evaluation metric. The AUC score can still be high with many false positive points given a large number of true positive points since low-valued false positive points do not affect the AUC score. However, all the false positive points decrease the normalized saliency value. Thus the NSS score penalizes all the false positive points.

Information Gain (IG) [49, 50]. is used for measuring saliency results beyond systematic bias.

Similarity (SIM) [76, 82]. measures the intersection between the predicted and the ground-truth saliency maps indicating the overlapping of the two maps. It is defined as the sum of the minimum value of the normalized predicted saliency map and the normalized ground-truth map. The similarity score is lower for sparse maps. It is sensitive to failed detection of saliency points since missing saliency values would lead to zero similarity, thus reducing the similarity score.

Pearson's Correlation Coefficient (CC) [52]. is a measurement for evaluating the correlation or dependence between the predicted and the ground-truth saliency maps.

Kullback-Leibler (KL) Divergence [40]. measures the difference between the distributions of the saliency map prediction and the ground truth, while other metrics mentioned above measure the similarity.

Computation details are given in *Supplementary Materials*. Together, these metrics have different properties regarding their sensitivity to false positives or false negatives, measurement, and metric categories:

Sensitivity: All the metrics are sensitive to false negatives, while KL, IG, and SIM penalize significantly for false negatives, especially when the predicted values are close to zero. The normalization step of NSS increases the penalty for detecting false positives and thus makes it more sensitive to false positives than other metrics. CC is a symmetric metric according to its definition, so it has equal sensitivity to false positives and false negatives. The AUC score is insensitive to false positives since it can still be high if the resulting saliency maps have many true positives.

Measurement: KL is a measurement for dissimilarity while other metrics are for similarity. Thus, better models have lower scores for KL and higher scores for other metrics.

Metric Category: Location-based metrics (AUC, NSS, IG) evaluate models based on fixation points, and distribution-based metrics (SIM, CC, KL) compute evaluation based on saliency maps as the continuous distribution.

5.2 Results

To set a benchmark for saliency map prediction, we compare computational saliency map models qualitatively and quantitatively and the predicted location bias. We use the first 52 image blocks (1872 images) in the dataset as the training data and the remaining 3 blocks of images (108 images) for testing. All the results we show here are evaluated on the test data.

5.2.1 Qualitative Evaluation. We show the qualitative comparison of different models for different UI types in Figure 8. For all the models, false positive errors are the main errors in the results. All the models can capture informative areas such as images and texts, but not all of them really attract the user's attention, and sometimes only a small part of an image is considered salient. Thus, it is generally challenging for predictive models to separate between informative areas and salient areas. GBVS and the pretrained UMSI typically capture all the image and text areas leading to high false positive error. Models trained on UEyes achieve better results compared to the pretrained models. Our improved model UMSI++ generates the closest saliency maps to the ground-truth fixations, compared to other models and across all the UI types.

5.2.2 Quantitative Comparison Across UI Types. We first show the importance of training on multiple types of UIs because training on one type leads to a drop in accuracy on other types. We train our UMSI++ model on either mobile UI or webpage training data in UEyes, respectively, and testing on all UI types on the same test set. The accuracy when predicting other UI types (different from those seen during training) drops significantly. For example, when the model was trained on mobile UIs, its accuracy decreased from 0.899 to 0.844 when tested on webpages, from 0.890 to 0.803 on desktop UIs, and from 0.924 to 0.849 on posters. Similarly, when the model was trained on webpages, its accuracy decreased from 0.905 to 0.832 on mobile UIs, from 0.890 to 0.813 on desktop UIs, and from 0.924 to 0.848 on posters. Thus, how people perceive visual hierarchies and look at UIs while viewing one type of UIs could not be really generalized to other UI types.

We quantitatively compare the models evaluated on the metrics mentioned in Section 5.1. We show that by training on UEyes, both the SAM and the UMSI models achieve higher accuracy and better generalization ability. Compared to other models, our improved model UMSI++ outperforms the state-of-the-art models on most metrics, as shown in Table 1. Since AUC is a standard evaluation metric ranging between 0 and 1 for saliency map prediction (larger values indicate higher accuracy), we quantitatively evaluate and compare different models across UI types as shown in Figure 9. The pretrained SAM model performs better than the pretrained UMSI. However, after training on UEyes, they perform similarly on all the UI types. By introducing new loss terms, UMSI++ achieved the best performance across all the UI types, while SAM++ does not have higher accuracy than the original SAM training on UEyes. For both SAM and UMSI architectures, desktops have the lowest accuracy among all the UI types aligning with our observation in the qualitative results.

5.2.3 Predicted Location Bias. We further visualize the location bias of saliency maps predicted by different models in Figure 10. All the models except GBVS can capture the top-left location bias of UIs. The models trained on our UEyes

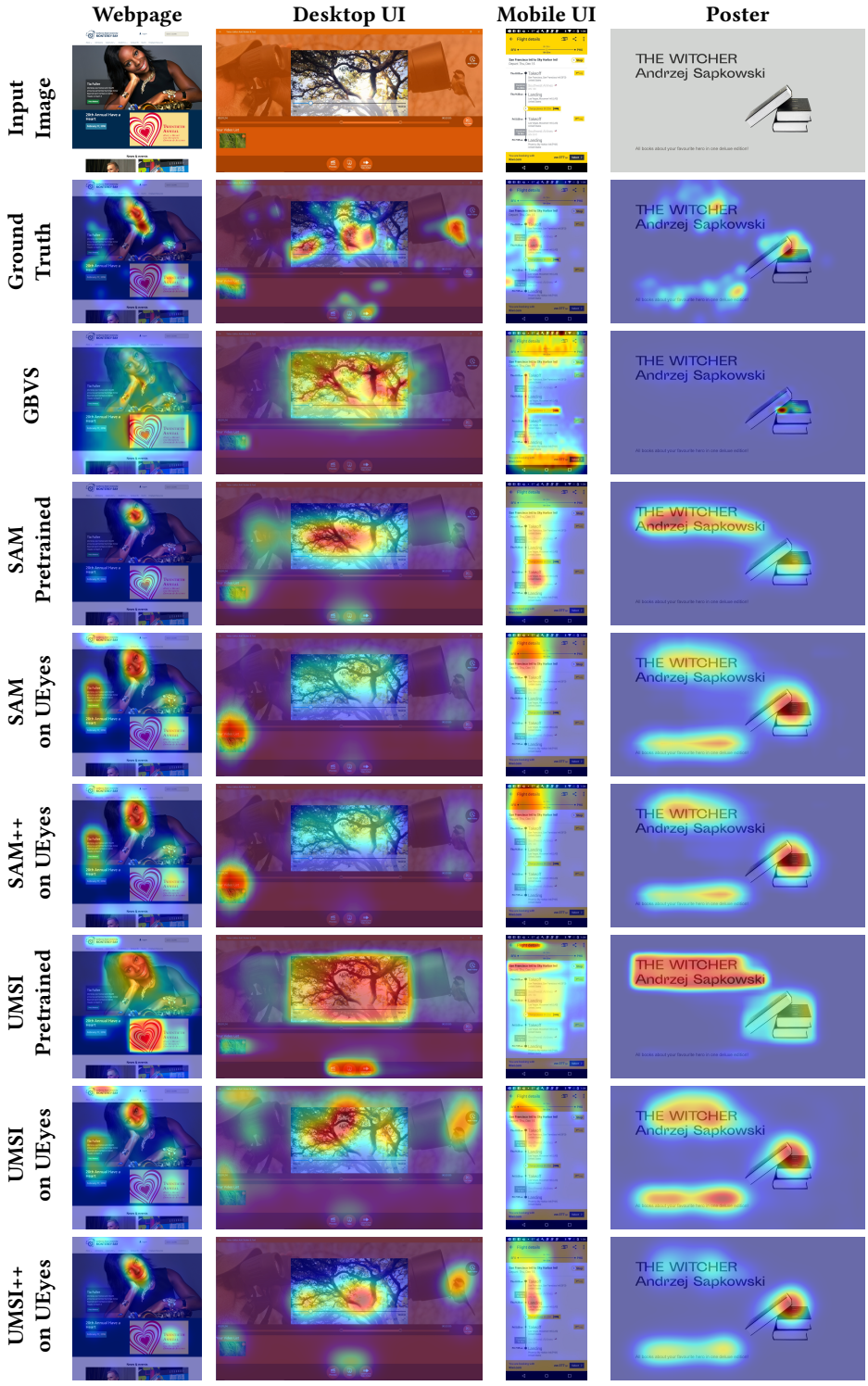


Fig. 8. Saliency maps qualitative comparison. Compared to other models, our improved model UMSI++ generates the closest saliency maps to the ground-truth across all the UI types.

Model	AUC-Judd ↑	NSS ↑	IG ↑	SIM ↑	CC ↑	KL ↓
GBVS	0.756 ± 0.104	0.256 ± 0.197	3.214 ± 0.668	0.513 ± 0.097	0.314 ± 0.193	3.916 ± 2.630
SAM Pretrained	0.822 ± 0.074	0.377 ± 0.170	3.143 ± 0.768	0.562 ± 0.081	0.522 ± 0.146	2.721 ± 1.457
SAM on UEyes	0.885 ± 0.057	**0.434 ± 0.185**	3.337 ± 0.774	0.663 ± 0.081	0.720 ± 0.127	2.016 ± 1.263
SAM++ on UEyes	0.868 ± 0.060	0.414 ± 0.179	3.165 ± 0.774	0.666 ± 0.080	0.717 ± 0.127	1.604 ± 1.185
UMSI Pretrained	0.778 ± 0.090	0.346 ± 0.178	3.177 ± 0.796	0.521 ± 0.078	0.431 ± 0.155	3.757 ± 1.769
UMSI on UEyes	0.878 ± 0.066	0.424 ± 0.187	**3.376 ± 0.807**	0.639 ± 0.085	0.699 ± 0.156	2.676 ± 1.408
UMSI++ on UEyes	**0.905 ± 0.044**	0.401 ± 0.173	3.320 ± 0.744	**0.733 ± 0.069**	**0.833 ± 0.078**	**1.166 ± 0.772**

Table 1. Saliency maps quantitative evaluation, reporting Mean ± SD for each metric. Arrows denote the direction of the importance, e.g., ↑ means 'higher is better'. We highlight the best column-wise result in bold typeface. UMSI++ outperforms other models on most evaluation metrics.

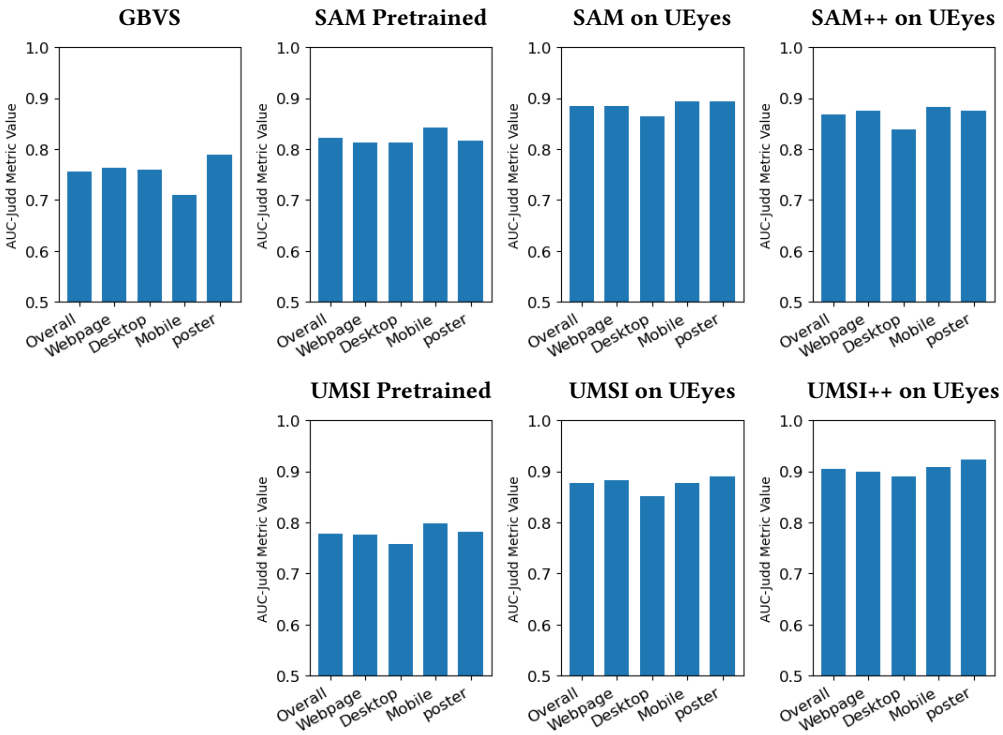


Fig. 9. Comparison of predictive accuracy of saliency map models using AUC-Judd (a measure of saliency map performance) as the metric. Larger values indicate higher accuracy. The figure shows that UMSI++ has the best performance across all the UI types.

dataset can capture more accurate location bias than the pretrained models. After training on UEyes, SAM, SAM++, and UMSI achieve similar saliency location bias. Our improved model UMSI++ has the highest similarity with the ground-truth location bias. From the visualization, we can also clearly see that all the models over-capture the salient areas and have many false positive errors, which aligns with what we found in the qualitative comparison. Saliency map prediction on webpages and especially desktop UIs are tough to detect. Salient areas on webpages are sparsely spread in the top-left quadrant than other types. The salient areas on desktop UIs are separated into two sub-areas, one

right above the center and the other around the top-left corner. It is much more challenging for the models to simulate sparser areas. The models trained on UEyes show better performance in capturing such sparse salient areas, while other models can only capture the entire areas with much more false positives. All the models except GBVS can well capture the most salient areas of mobile UIs (the top-left quadrant) and posters (right above the center). The models trained on UEyes show similar accurate location bias results on webpages, desktop UIs, and mobile UIs. However, UMSI++ reveals better location bias on posters than other models by detecting decreased salient at the top of posters.

6 ASSESSING SCANPATH MODELS

In scanpath prediction, the goal is to predict a sequence of fixations. The problem is much more challenging than saliency maps because the order of fixations must be retained. We report results on how well computational models fare with the different UI types. We compared four models:

The Itti-Koch-based model [37]. is a model proposed in the pre-deep learning era. It first generates a saliency map by extracting visual features for intensity, color, and orientation by a set of linear center-surround operations. It then takes the "winner-take-all" strategy to select the attended position. It repeatedly applies the "inhibition of return" feedback to inhibit the chosen position in the saliency map to get the resulting scanpath.

DeepGaze III [47]. predicts the sequence of fixation points in scanpaths over static images. It takes both the input image and the positions of the previous four fixation points to predict the density/probabilistic map for the next fixation point. It then generates the scanpath by recursively selecting the next fixation point with the highest probability value on the density map and adding the new predicted fixation point to infer the density map for the next point. Compared to PathGAN, DeepGaze III is a method concentrated on fixation point detection to form the final scanpath.

DeepGaze++ (Ours). Although DeepGaze III can take the information of previous fixation points to generate the density of the next point, it often leads to similar density maps for consecutive fixation point prediction. We propose modification where we repeatedly select the position with the highest probability on the density map and apply "inhibition of return" to inhibit the chosen position in the saliency map. For the i-th previous fixation point information, we assign a weight of $1 - 0.1 \cdot (i - 1)$ to the "inhibition of return" feedback so that older fixation points have less effect on the prediction results.

PathGAN [3]. is a deep convolutional-recurrent neural network trained on adversarial examples. The generator takes the image as input to generate the corresponding scanpath. The discriminator encodes both the image and the scanpath to discriminate whether a scanpath is realistic for a given image. Thus, PathGAN can generate more realistic scanpaths. However, PathGAN only focuses on the path and cannot predict fixation points.

PathGAN++ (Ours). Since the PathGAN model can generate more accurate trajectories for scanpaths than other models, we introduce a Dynamic Time Warping (DTW) loss term that maximizes the similarity between the predicted scanpath and the ground truth in temporal order to increase the accuracy of scanpaths further.

6.1 Evaluation Metrics

We used six metrics with different properties to evaluate scanpath models. They are commonly used metrics for scanpath evaluation [1, 28]. The first three metrics described below (DTW, TDE, Eyenalysis) are the most commonly used in

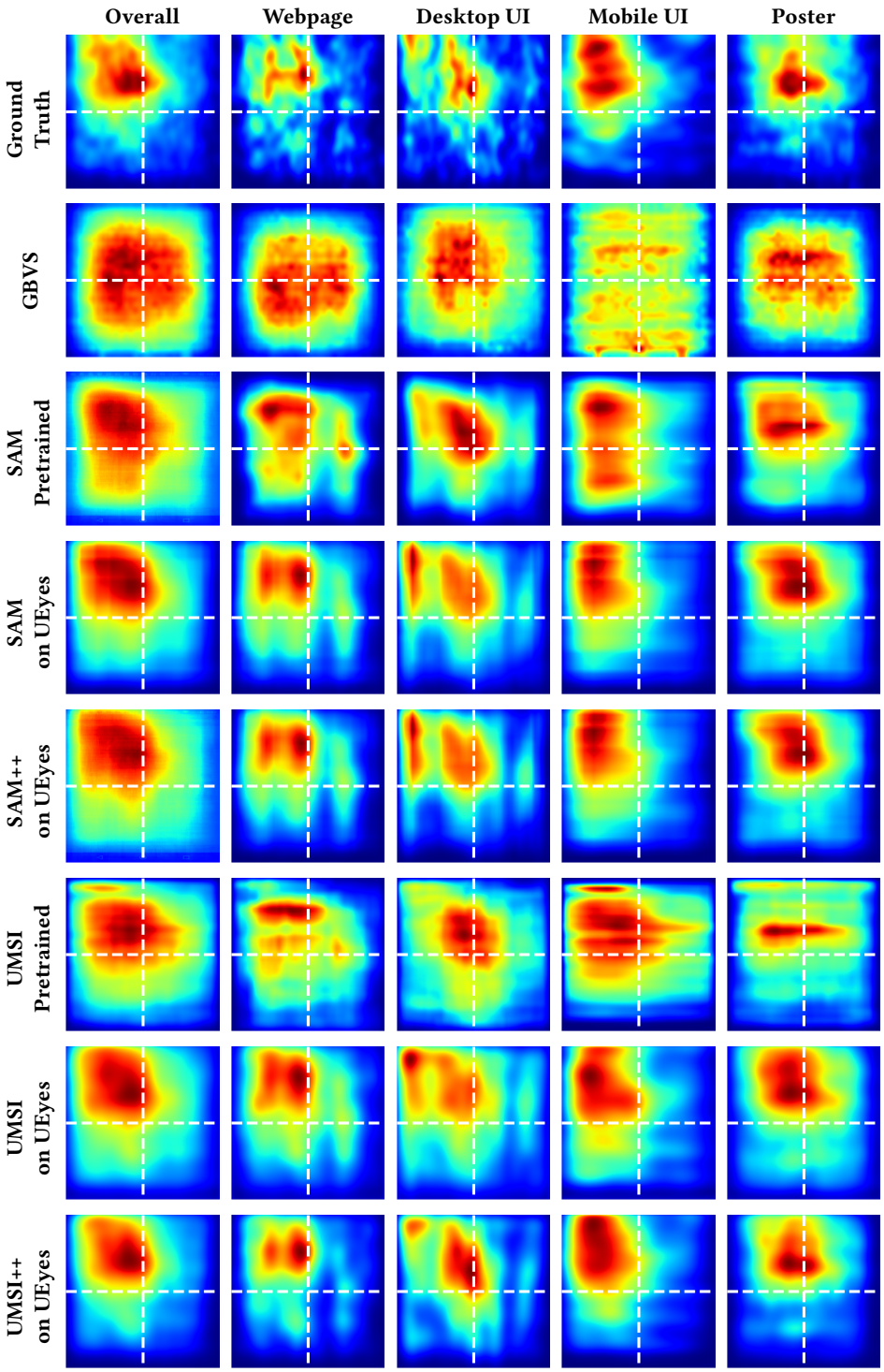


Fig. 10. Comparison of the location bias of saliency maps predicted by different models across UI types. UMSI++ has the highest similarity with the ground-truth location bias.

evaluating scanpaths, as they capture the temporal and spatial aspects of visual attention. We include three additional metrics for completeness.

Dynamic Time Warping (DTW). is a standard metric for measuring similarity between two temporal sequences with different lengths [5, 77]. It finds the optimal match and computes the distance between them for two scanpaths monotonically without missing essential features.

Time Delay Embedding (TDE). creates the sets of time-delay embedding vectors for the predicted and the ground truth scanpaths, respectively, by collecting all the consecutive subscanpaths of a given length as vectors [84, 88]. We look for the vector from the predicted scanpath for each time-delay embedding vector from the ground truth scanpath with the minimum distance. Thus, TDE measures the differences between subscanpaths to evaluate the scanpaths.

Eyenalysis. performs a double mapping between two scanpaths [60]: For each fixation point of one scanpath, we find the spatially closest fixation point of the other scanpath. We repeat the same procedure the other way around. Eyenalysis measures the average distances for all the closest pairs found.

Cross-Recurrence (REC). measures the matching ratio of fixation points in the two scanpaths [98]. We first truncate the two scanpaths to the same length, which is the minimum length of the two scanpaths. We define fixation pairs that have distance within a certain threshold as recurrences (we set the threshold to be the image size scaled by 0.05). REC counts the recurrences and computes the percentage of recurrences out of all the fixation pairs in the two scanpaths.

Weighted Determinism (DET). is the percentage of recurrent fixation points on subscanpaths in which all the pairs of corresponding fixation points are recurrences, and all such recurrent fixation point pairs contain different fixation points from both scanpaths [1, 28]. The original Determinism measure [1] only counts the number of corresponding subscanpaths. We propose to compute the percentage of fixation points to measure the subscanpaths better.

Center of Recurrence Mass (CORM). measures the distance between the center of recurrences indicating the dominant lag of recurrences [1, 28]. The CORM score is smaller when the recurrent fixation point pairs in the scanpaths occur close in time.

DTW, TDE, and Eyenalysis measure fixation position and sequence in the temporal order as they match the two sequences differently. REC and DET only measure the similarity of fixation positions. They have higher values if the fixation points in the two sequences are close, regardless of the temporal order. The CORM is a measurement for detecting the dominant lag of recurrences.

6.2 Results

6.2.1 Qualitative Evaluation. We show the qualitative comparisons of different models across UI types in Figure 11. Overall, all the models cannot predict accurate results compared to the ground-truth data. The pretrained PathGAN model and the DeepGaze III model get stuck in local areas so that the predicted points are clustered. Due to the similar density maps predicted by DeepGaze III for consecutive fixation point prediction, the DeepGaze III model selects very close positions for fixation points leading to a cluster of points and getting stuck in that cluster. PathGAN can only generate the scanpath without considering fixation points. Thus, it is impossible to infer which points users pay more visual attention to based on the PathGAN results. The Itti-Koch-based model, PathGAN trained on UEyes, DeepGaze++, and PathGAN++ show better prediction results. However, most scanpaths predicted by PathGAN and PathGAN++

trained on UEyes are biased to be around the center of the UIs. All models tend to predict scanpaths with many fixation points, and not on salient areas.

6.2.2 Quantitative Evaluation. In Table 2, we compare different models based on the mentioned evaluation metrics in Section 6.1. For a fair comparison, since these metrics are dependent on the scanpath length, we make sure that predicted scanpaths generated by all the models have exactly 15 fixation points. Since DTW is a standard evaluation metric for scanpaths (smaller values indicate higher accuracy), we further visualize the comparison of models on the DTW metric across UI types as shown in Figure 12. Desktop UIs have higher DTW value, i.e., lower accuracy than other UI types predicted by all the models except the Itti-Koch-based model, indicating that scanpaths on desktop UIs are harder to be predicted. All models perform best on mobile UIs. DeepGaze III exhibited the worst performance among all the models for UI types except mobile UIs. PathGAN++ model shows superior performance compared to other models in terms of DTW, TDE, and Eyenalysis. This shows that our model can simulate real scanpath trajectories. However, the results are still qualitatively inaccurate. This shows that the current metrics for evaluating scanpaths may not be sufficient to capture the more nuanced details of eye movements.

6.2.3 Comparison Between PathGAN++ and DeepGaze++. Each model has its own strengths and limitations when comparing the performance of PathGAN++ and DeepGaze++. PathGAN++ excels at generating realistic trajectories thanks to its discriminative component in the model arhictecture. However, it falls short in predicting proper fixation points, and the generated points are often outside the areas of interest. On the other hand, DeepGaze++ is better at predicting fixation points, as it is based on saliency maps that highlight elements on the UIs. However, it can suffer from repetitive density maps for consecutive fixation point predictions, leading to unrealistic scanpaths. Additionally, the "inhibition of return" of the model is deterministic and not differentiable, so it cannot be optimized by any loss terms, which can hinder its optimization.

Model	DTW ↓	TDE ↓	Eyenalysis ↓	REC ↑	DET ↑	CORM ↓
Itti-Koch-based	6.282 ± 0.973	0.147 ± 0.027	0.043 ± 0.022	2.224 ± 2.053	2.021 ± 10.854	34.497 ± 22.890
DeepGaze III Pretrained	7.650 ± 2.899	0.250 ± 0.078	0.124 ± 0.072	1.290 ± 3.281	1.025 ± 8.510	**13.838 ± 24.082**
DeepGaze++	5.230 ± 1.180	0.133 ± 0.031	**0.043 ± 0.022**	1.876 ± 1.700	1.778 ± 10.046	31.590 ± 23.120
PathGAN Pretrained	4.381 ± 1.559	0.160 ± 0.054	0.072 ± 0.036	**3.896 ± 5.049**	**7.039 ± 18.651**	22.528 ± 22.970
PathGAN on UEyes	4.354 ± 1.322	0.121 ± 0.040	0.045 ± 0.024	2.414 ± 2.455	5.687 ± 17.960	27.613 ± 21.644
PathGAN++	**4.236 ± 1.332**	**0.120 ± 0.041**	**0.043 ± 0.022**	2.810 ± 2.743	5.761 ± 16.053	27.956 ± 21.544

Table 2. Scanpaths evaluation, reporting Mean ± SD for each metric. Arrows denote the direction of the importance, e.g., ↑ means 'higher is better'. We highlight the best column-wise result in bold typeface. PathGAN++ outperforms other models on all three evaluation metrics measuring the fixation sequence in the temporal order (DTW, TDE, Eyenalysis).

6.2.4 Saccade Angle and Amplitude Distribution. We show the saccade angle and amplitude distribution comparison for models in Figure 13. All the models cannot capture the same distributions as the ground-truth data. Human saccade directions are primarily from left to right, along with a small portion of top to bottom. The pretrained PathGAN model and DeepGaze III have clustered distributions due to the stuck points on the predicted scanpaths. PathGAN trained on UEyes and PathGAN++ has an incorrect center bias on the distribution. Further, the inhibition of return implemented in the Itti-Koch-based model and DeepGaze++ avoids small distances between consecutive fixation points. Thus, the saccade amplitudes are more significant than the ground truth. We found that most saccade directions predicted by

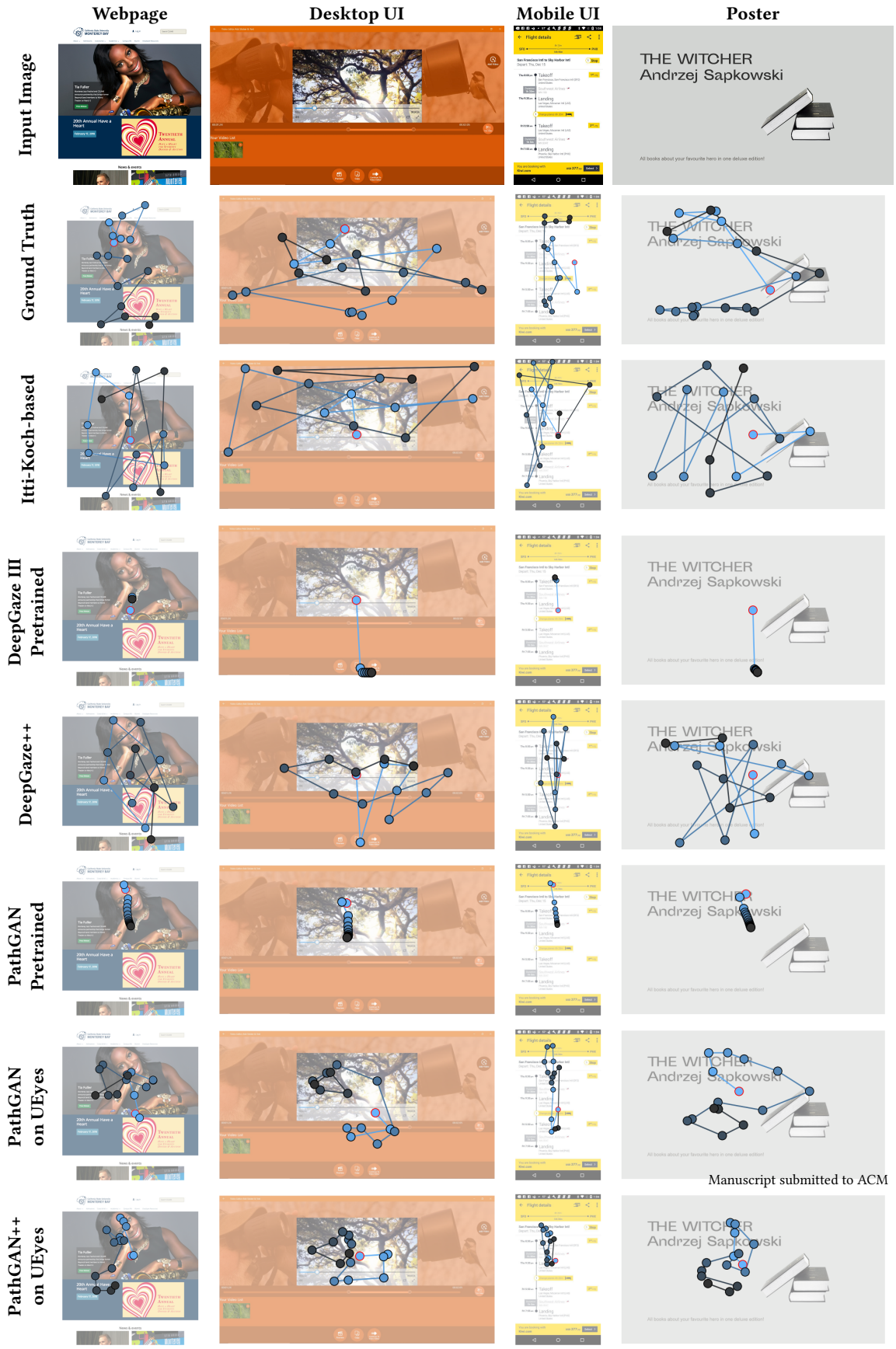


Fig. 11. Scanpath qualitative comparison. DeepGaze++ can better predict fixation points but cannot predict realistic scanpaths. PathGAN++ can predict the realistic trajectory but not the accurate fixation points. Trajectories begin with a blue color and end with a black color. The starting point is highlighted with a red border.

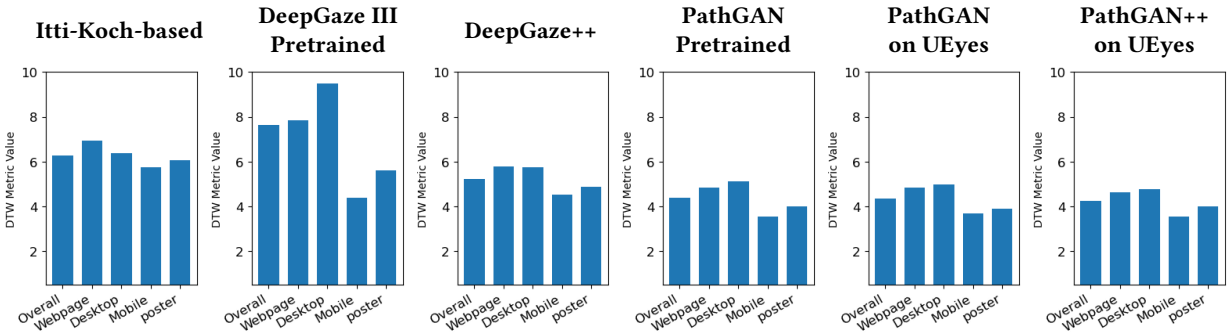


Fig. 12. Comparison of predictive accuracy of scanpath models using DTW (a measure for the fixation sequence in the temporal order) as the metric. Smaller values indicate higher accuracy. The figure shows that DeepGaze++ has the best performance across all the UI types.

DeepGaze++ are towards the right for desktop UIs, mobile UIs, and posters, which shows that DeepGaze++ can capture the correct tendencies on these UI types. However, it is still wrong with webpages and cannot predict the tendency to the bottom of the UIs.

6.2.5 Visited and Revisited Elements. We show the visited and revisited element ratio comparison for the models in *Supplementary Materials*. All the models can correctly predict that text elements are more likely to be fixated than images. The pretrained PathGAN model and DeepGaze III underestimate the visiting and revisiting ratios. DeepGaze++ has the best visiting ratio prediction but overestimates the revisiting ratios for all UI types. All the models capture that mobile UIs have lower visiting and revisiting ratios than other UI types. Still, all the models except DeepGaze++ underestimate the visiting and revisiting ratios for mobile UIs. Most models predict the closest visiting and revisiting element ratios on posters to the ground truth. PathGAN++ has the closest prediction for visiting and revisiting element ratios for most UI types except underestimating mobile UIs.

7 DISCUSSION

This paper has shed new light on eye movement behavior on different UI types. We summarize the main findings on how people look at UIs and address challenges and limitations with current computational models.

7.1 How People Look at UIs

We have found that, in general, users pay more attention to the top-left area of a UI. While this was found earlier for mobile UIs [53] have can now confirm a similar trend across all types of UIs considered in this work, including poster designs. We also have found that eye saccade directions are mainly directed towards the right or bottom part of the UI. Further, directions towards the right have larger distances between consecutive points than those towards the bottom.

On the other hand, text elements are more likely to be fixated on than images, which explains why saccade directions tend to go from left to right rather than vice versa; although this may be an artifact of our dataset, since most of our UIs are in English and thus, participants were enforced to read text from left to right. Still, the ratio of images to texts does not affect the ratios of visited and revisited elements in these two element categories. In addition, saccades towards the right part of the UIs have larger distances between consecutive points than those towards the bottom. These findings

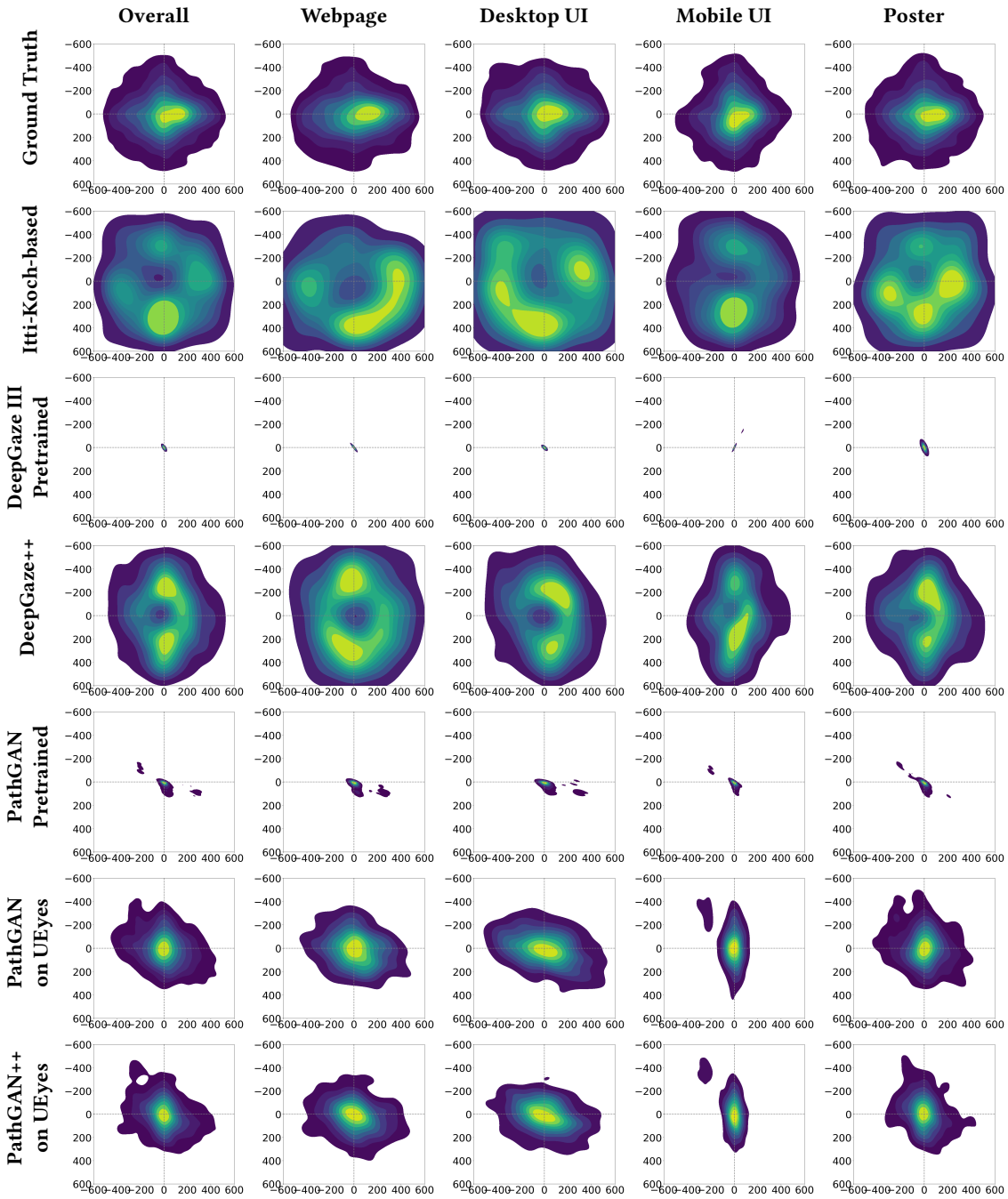


Fig. 13. Saccade angle and amplitude distribution comparison for scanpath predictive models. Human saccade directions are primarily from left to right, along with a small portion from top to bottom. All models cannot capture the same distributions as the ground-truth data.

support the idea that user interfaces are not glanced at like natural scenes [53]. Instead of a center bias, there is a strong top-left bias.

Our data allows us to dive deeper into subtle differences among different types of UIs:

Webpage: We found that users tend to look from left to right on webpages with larger inter-fixation distances. These large distances might explain why computational scanpath models exhibit their worst performance with webpages, while computational saliency models perform so well with other types of UIs.

Desktop UI: There are two salient areas in Desktop UIs: one right above the center and the top-left corner. This makes it difficult for computational models of saliency maps and scanpaths to deliver accurate predictions, which were found to perform poorly with such multi-modal gaze distributions.

Mobile UI: Mobile UIs have lower visit and revisit ratios than other UI types, indicating that users focus more on the "attractive" elements while ignoring the others, and have a lower tendency to go back to look at the same content. Further, we noticed that most scanpath models can predict the low visit ratio of mobile UIs, and thus have the best predictive accuracy on mobile UIs compared to other types.

Poster: As with other UIs, users tend to scan posters from left to right, with a small number of directions towards the bottom with various fixation distances. The distances of the consecutive fixation points have a more considerable variation than other UI types, which is harder to predict by the current computational scanpath models.

7.2 Current Computational Models

Our results highlight that, when predicting visual saliency, training computational models with eye movement on user interfaces yields superior performance to training with proxy data, like mouse movements or manual annotations, or even training with data collected on natural scenes. This is unsurprising, but our results demonstrate its superiority quantitatively. In particular, we showed that training UMSI on UEyes increases its AUC performance from 0.778 to 0.878. We inspected the predictions and found that much of this difference can be attributed to over-detection cases by UMSI: it predicts saliency across more extensive areas of the UI than a user may have had time to inspect; and this is reflected in its high false positive rate. However, after training on our dataset and with our model modifications, the accuracy of UMSI improved considerably.

7.3 Limitation and Future Work

7.3.1 Mobile UI Viewing Setting. The fixed screen used in our experiments ensures consistent data collection and analysis across different UI types. However, this setting does not accurately simulate the real-life mobile UI viewing experience while holding mobile phones. To improve the realism in this regard, one could rescale the size of the mobile UI screenshots based on a mean viewing distance of 30 cm, as recommended by previous studies [53, 57]. Considering that the distance between participants' eyes and the screen is around 60 cm in our experiment, the physical size of the displayed stimuli should be roughly twice larger as it is on a mobile screen. However, our findings on mobile UIs corroborate previous findings by Leiva et al [53].

7.3.2 Semantic Understanding of UI Elements. The current classification of visited and revisited UI elements into broad categories (text, images, face) does not capture the semantic differences within each category. Future work can focus on developing more detailed and nuanced classifications of visited and revisited UI elements by extracting their semantic meaning [93] to gain a more accurate understanding of user gaze behaviors.

7.3.3 False Positives on Saliency maps. While computational models of saliency maps can capture informative areas such as images and texts, they still tend to generate false positive errors and over-detect salient areas, leading to a lack of accuracy and reliability. Future work can improve the model's ability to differentiate between truly salient areas and false positives and use additional features, such as user task goals, to guide the saliency prediction.

7.3.4 Inaccurate Scanpath Models. Current scanpath models cannot accurately capture both the scanpath trajectories and fixation points of human eye movements with the same model. Further improving the model requires a better understanding of the factors influencing gaze behaviors, such as visit and revisit tendencies, and incorporating these factors into the model. Additionally, better evaluation metrics are needed to assess the quality of predicted scanpaths. Developing such metrics will help better understand the performance of scanpath models and guide the design of improved models in the future.

7.3.5 Individual Differences. Individuals have different viewing strategies when looking at user interfaces. These differences in viewing strategies can impact gaze behavior, and they need to be accounted for in predictive models. Future work can focus on understanding and modeling the individual differences in viewing strategies across different UI types, which can be achieved by developing personalized, predictive models incorporating individual differences.

8 CONCLUSION

In this paper, we present UEyes, a large-scale eye-tracking dataset that includes 1980 UIs in various types, along with multi-duration saliency maps and scanpaths. Moreover, we present the first in-depth analysis and comparison of eye movement tendencies across common UI types. We report the performance of existing predictive models for saliency maps and scanpaths across the UI types.

Open Science

The dataset and trained models are available at [URL TBA]. The dataset includes raw CSV log files recorded with the GP3 HD eye tracker, associated heatmaps and scanpaths, image stimuli (screenshots), and metadata about the design type.

REFERENCES

[1] Nicola C Anderson, Fraser Anderson, Alan Kingstone, and Walter F Bischof. 2015. A comparison of scanpath comparison methods. *Behavior research methods* 47, 4 (2015), 1377–1392.

[2] Marc Assens, Xavier Giro-i Nieto, Kevin McGuinness, and Noel E. O'Connor. 2017. SaltiNet: Scan-Path Prediction on 360 Degree Images Using Saliency Volumes. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. 2331–2338. https://doi.org/10.1109/ICCVW.2017.275

[3] Marc Assens, Xavier Giro i Nieto, Kevin McGuinness, and Noel E. O'Connor. 2018. PathGAN: Visual Scanpath Prediction with Generative Adversarial Networks. *ECCV Workshop on Egocentric Perception, Interaction and Computing (EPIC)*.

[4] Roman Bednarik and Markku Tukiainen. 2007. Validating the restricted focus viewer: A study using eye-movement tracking. *Behavior research methods* 39, 2 (2007).

[5] Donald J Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series.. In *KDD workshop*, Vol. 10. Seattle, WA, USA:, 359–370.

[6] Sergey Bezryadin, Pavel Bourov, and Dmitry Ilinih. 2007. Brightness Calculation in Digital Image Processing. In *Proc. TDPF Symposium*.

[7] Ali Borji. 2019. Saliency Prediction in the Deep Learning Era: Successes, Limitations, and Future Challenges. In *CoRR abs/1810.03716 (arXiv preprint)*.

[8] A. Borji and L. Itti. 2013. State-of-the-art in visual attention modeling. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1 (2013).

[9] Ali Borji and Laurent Itti. 2015. CAT2000: A Large Scale Fixation Dataset for Boosting Saliency Research. *CVPR 2015 workshop on "Future of Datasets"* (2015). arXiv preprint arXiv:1505.03581.

[10] Ali Borji, Hamed R. Tavakoli, Dicky N. Sihite, and Laurent Itti. 2013. Analysis of Scores, Datasets, and Models in Visual Saliency Prediction. In *Proc. ICCV*.

[11] Michelle A Borkin, Zoya Bylinskii, Nam Wook Kim, Constance May Bainbridge, Chelsea S Yeh, Daniel Borkin, Hanspeter Pfister, and Aude Oliva. 2015. Beyond memorability: Visualization recognition and recall. *IEEE transactions on visualization and computer graphics* 22, 1 (2015).

[12] Maximilian D Broda and Benjamin de Haas. 2022. Individual fixation tendencies in person viewing generalize from images to videos. *i-Perception* 13, 6 (2022), 20416695221128844.

[13] Neil Bruce and John Tsotsos. 2005. Saliency based on information maximization. *Advances in neural information processing systems* 18 (2005).

[14] Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. 2015. Mit saliency benchmark. (2015).

[15] Zoya Bylinskii, Nam Wook Kim, Peter O'Donovan, Sami Alsheikh, Spandan Madan, Hanspeter Pfister, Fredo Durand, Bryan Russell, and Aaron Hertzmann. 2017. Learning Visual Importance for Graphic Designs and Data Visualizations. In *Proc. UIST*.

[16] Ying Cao, Rynson WH Lau, and Antoni B Chan. 2014. Look over here: Attention-directing composition of manga elements. *ACM Transactions on Graphics (TOG)* 33, 4 (2014).

[17] Zhaohui Che, Ali Borji, Guangtao Zhai, Xiongkuo Min, Guodong Guo, and Patrick Le Callet. 2020. How is Gaze Influenced by Image Transformations? Dataset and Model. *IEEE Transactions on Image Processing* 29 (2020), 2287–2300. https://doi.org/10.1109/TIP.2019.2945857

[18] Zhenzhong Chen and Wanjie Sun. 2018. Scanpath Prediction for Visual Attention Using IOR-ROI LSTM *(IJCAI'18)*. AAAI Press, 642–648.

[19] L. Cooke. 2006. Is the mouse a poor man's eye tracker?. In *Proc. STC*.

[20] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. 2016. A Deep Multi-Level Network for Saliency Prediction. In *International Conference on Pattern Recognition (ICPR)*.

[21] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. 2018. Predicting Human Eye Fixations via an LSTM-Based Saliency Attentive Model. *IEEE Transactions on Image Processing* 27, 10 (2018). https://doi.org/10.1109/TIP.2018.2851672

[22] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. 2018. SAM: Pushing the Limits of Saliency Prediction Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision and Pattern Recognition Workshops*.

[23] Desktop UI Dataset. 2020. . https://github.com/waltteri/desktop-ui-dataset

[24] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. Rico: A Mobile App Dataset for Building Data-Driven Design Applications. In *Proceedings of the 30th Annual Symposium on User Interface Software and Technology (UIST '17)*.

[25] Guanqun Ding, Nevrez İmamoğlu, Ali Caglayan, Masahiro Murakawa, and Ryosuke Nakamura. 2022. SalFBNet: Learning pseudo-saliency distribution via feedback convolutional networks. *Image and Vision Computing* 120 (2022), 104395. https://doi.org/10.1016/j.imavis.2022.104395

[26] Richard Droste, Jianbo Jiao, and J. Alison Noble. 2020. Unified Image and Video Saliency Modeling. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 419–435.

[27] Sergio Etchebehere and Elena Fedorovskaya. 2017. On the Role of Color in Visual Saliency. *Intl. Symp. Electronic Imaging* 6 (2017).

[28] Ramin Fahimi and Neil DB Bruce. 2021. On metrics for measuring scanpath similarity. *Behavior Research Methods* 53, 2 (2021), 609–628.

[29] Camilo Fosco, Vincent Casser, Amish Kumar Bedi, Peter O'Donovan, Aaron Hertzmann, and Zoya Bylinskii. 2020. Predicting visual importance across graphic design types. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 249–260.

[30] Camilo Fosco, Anelise Newman, Pat Sukhum, Yun Bin Zhang, Nanxuan Zhao, Aude Oliva, and Zoya Bylinskii. 2020. How much time do you have? modeling multi-duration saliency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4473–4482.

[31] S. Frintrop, E. Rome, and H. I. Christensen. 2010. Computational visual attention systems and their cognitive foundations: A survey. *ACM Trans. Appl. Percept.* 7, 1 (2010).

[32] Shahrbanoo Hamel, Nathalie Guyader, Denis Pellerin, and Dominique Houzet. 2014. Contribution of Color Information in Visual Saliency Model for Videos. In *Proc. ICISP*. 213–221.

[33] Rui Han and Shuangjiu Xiao. 2018. Human Visual Scanpath Prediction Based on RGB-D Saliency. In *Proceedings of the 2018 International Conference on Image and Graphics Processing* (Hong Kong, Hong Kong) *(ICIGP 2018)*. Association for Computing Machinery, New York, NY, USA, 180–184. https://doi.org/10.1145/3191442.3191463

[34] Jonathan Harel, Christof Koch, and Pietro Perona. 2006. Graph-based visual saliency. *Advances in neural information processing systems* 19 (2006).

[35] J. M. Henderson. 1993. Eye movement control during visual object processing: effects of initial fixation position and semantic constraint. *Can. J. Exp. Psychol.* 47, 1 (1993).

[36] Xun Huang, Chengyao Shen, Xavier Boix, and Qi Zhao. 2015. SALICON: Reducing the Semantic Gap in Saliency Prediction by Adapting Deep Neural Networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*. 262–270. https://doi.org/10.1109/ICCV.2015.38

[37] Laurent Itti, Christof Koch, and Ernst Niebur. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence* 20, 11 (1998), 1254–1259.

[38] Sen Jia. 2018. EML-NET: An Expandable Multi-Layer NETwork for Saliency Prediction. *CoRR* abs/1805.01047 (2018). arXiv:1805.01047 http://arxiv.org/abs/1805.01047

[39] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. 2015. SALICON: Saliency in Context. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1072–1080. https://doi.org/10.1109/CVPR.2015.7298710

[40] James M Joyce. 2011. Kullback-leibler divergence. In *International encyclopedia of statistical science*. Springer, 720–722.

[41] Tilke Judd, Frédo Durand, and Antonio Torralba. 2012. A Benchmark of Computational Models of Saliency to Predict Human Fixations. In *MIT Technical Report*.

[42] T. Judd, K. Ehinger, F. Durand, and A. Torralba. 2009. Learning to predict where humans look. In *Proc. ICCV*.

[43] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. 2009. Learning to predict where humans look. In *2009 IEEE 12th international conference on computer vision*. IEEE, 2106–2113.

[44] Nam Wook Kim, Zoya Bylinskii, Michelle A Borkin, Krzysztof Z Gajos, Aude Oliva, Frẽdo Durand, and Hanspeter Pfister. 2017. BubbleView: an interface for crowdsourcing image importance maps and tracking visual attention. *ACM Transactions on Computer-Human Interaction (TOCHI)* 24, 5 (2017). https://doi.org/10.1145/3131275

[45] Nam Wook Kim, Zoya Bylinskii, Michelle A Borkin, Aude Oliva, Krzysztof Z Gajos, and Hanspeter Pfister. 2015. A crowdsourced alternative to eye-tracking for visualization understanding. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. 1349–1354.

[46] Alexander Kroner, Mario Senden, Kurt Driessens, and Rainer Goebel. 2020. Contextual encoder–decoder network for visual saliency prediction. *Neural Networks* 129 (2020), 261–270. https://doi.org/10.1016/j.neunet.2020.05.004

[47] Matthias Kümmerer, Matthias Bethge, and Thomas SA Wallis. 2022. DeepGaze III: Modeling free-viewing human scanpaths with deep learning. *Journal of Vision* 22, 5 (2022).

[48] Matthias Kümmerer, Lucas Theis, and Matthias Bethge. 2014. Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. *arXiv preprint arXiv:1411.1045* (2014).

[49] Matthias Kümmerer, Thomas Wallis, and Matthias Bethge. 2014. How close are we to understanding image-based saliency? *arXiv preprint arXiv:1409.7686* (2014).

[50] Matthias Kümmerer, Thomas SA Wallis, and Matthias Bethge. 2015. Information-theoretic model comparison unifies saliency metrics. *Proceedings of the National Academy of Sciences* 112, 52 (2015), 16054–16059.

[51] Matthias Kummerer, Thomas SA Wallis, Leon A Gatys, and Matthias Bethge. 2017. Understanding low-and high-level contributions to fixation prediction. In *Proceedings of the IEEE international conference on computer vision*. 4789–4798.

[52] Olivier Le Meur, Patrick Le Callet, and Dominique Barba. 2007. Predicting visual fixations on video based on low-level visual features. *Vision research* 47, 19 (2007), 2483–2498.

[53] Luis A Leiva, Yunfei Xue, Avya Bansal, Hamed R Tavakoli, Tuðçe Köroðlu, Jingzhou Du, Niraj R Dayama, and Antti Oulasvirta. 2020. Understanding visual saliency in mobile user interfaces. In *22nd International conference on human-computer interaction with mobile devices and services*. 1–12.

[54] Guanbin Li and Yizhou Yu. 2015. Visual Saliency Based on Multiscale Deep Features. In *Proc. CVPR*. 5455–5463.

[55] Akis Linardos, Matthias Kümmerer, Ori Press, and Matthias Bethge. 2021. DeepGaze IIE: Calibrated prediction in and out-of-domain for state-of-the-art saliency modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 12919–12928.

[56] Gitte Lindgaard, Gary Fernandes, Cathy Dudek, and J. Brown. 2006. Attention web designers: You have 50 milliseconds to make a good first impression! *Behav. Inform. Technol.* 25, 2 (2006).

[57] Jennifer Long, Rene Cheung, Simon Duong, Rosemary Paynter, and Lisa Asper. 2017. Viewing distance and eyestrain symptoms with prolonged viewing of smartphones. *Clinical and Experimental Optometry* 100, 2 (2017), 133–137.

[58] Jianxun Lou, Hanhe Lin, David Marshall, Dietmar Saupe, and Hantao Liu. 2022. TranSalNet: Towards perceptually relevant visual saliency prediction. *Neurocomputing* 494 (2022), 455–467. https://doi.org/10.1016/j.neucom.2022.04.080

[59] S. Marat, A. Rahman, D. Pellerin, N. Guyader, and D. Houzet. 2013. Improving Visual Saliency by Adding 'Face Feature Map' and 'Center Bias'. *Cogn. Comput.* 5, 1 (2013).

[60] S Mathot, F Cristino, ID Gilchrist, and J Theeuwes. 2012. Eyenalysis: A similarity measure for eye movement patterns. *Journal of Eye Movement Research* 5 (2012), 1–15.

[61] Aliaksei Miniukovich and Antonella De Angeli. 2014. Visual Impressions of Mobile App Interfaces. In *Proc. NordiCHI*. 31–40.

[62] Aliaksei Miniukovich and Maurizio Marchese. 2020. Relationship between visual complexity and aesthetics of webpages. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[63] A. Mishra, Y. Aloimonos, and C.L. Fah. 2009. Active segmentation with fixation. In *Proc. ICCV*. 468–475.

[64] Thuyen Ngo and B.S. Manjunath. 2017. Saccade gaze prediction using a recurrent neural network. In *2017 IEEE International Conference on Image Processing (ICIP)*. 3435–3439. https://doi.org/10.1109/ICIP.2017.8296920

[65] A. Nuthmann and J. M. Henderson. 2014. Object-based attentional selection in scene viewing. *J. Vis.* 10, 8 (2014).

[66] J. P. Ossandon, S. Onat, and P. König. 2014. Spatial biases in viewing behavior. *J. Vis.* 14, 2 (2014).

[67] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2014. Learning layouts for single-pagegraphic designs. *IEEE transactions on visualization and computer graphics* 20, 8 (2014).

[68] Junting Pan, Cristian Canton, Kevin McGuinness, Noel E. O'Connor, Jordi Torres, Elisa Sayrol, and Xavier and Giro-i Nieto. 2017. SalGAN: Visual Saliency Prediction with Generative Adversarial Networks. In *arXiv*.

[69] Xufang Pang, Ying Cao, Rynson WH Lau, and Antoni B Chan. 2016. Directing user attention via visual flow on web designs. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–11.

[70] Robert J Peters, Asha Iyer, Laurent Itti, and Christof Koch. 2005. Components of bottom-up gaze allocation in natural images. *Vision research* 45, 18 (2005), 2397–2416.

[71] Hamed R. Tavakoli, Ali Borji, Jorma Laaksonen, and Esa Rahtu. 2017. Exploiting inter-image similarity and ensemble of extreme learners for fixation prediction using deep features. *Neurocomputing* 244 (2017), 10–18. https://doi.org/10.1016/j.neucom.2017.03.018

[72] Subramanian Ramanathan, Harish Katti, Nicu Sebe, Mohan Kankanhalli, and Tat-Seng Chua. 2010. An eye fixation database for saliency detection in images. In *European conference on computer vision*. Springer, 30–43.

[73] K. Rayner, S. P. Liversedge, A. Nuthmann, R. Kliegl, and Underwood G. 2009. Rayner's 1979 paper. *Perception* 38, 6 (2009).

[74] Hamed Rezazadegan Tavakoli, Esa Rahtu, and Janne Heikkilä. 2013. Stochastic bottom–up fixation prediction and saccade generation. *Image and Vision Computing* 31, 9 (2013), 686–693. https://doi.org/10.1016/j.imavis.2013.06.006

[75] Ruth Rosenholtz, Amal Dorai, and Rosalind Freeman. 2011. Do Predictions of Visual Perception Aid Design? *ACM Trans. Appl. Percept.* 8, 2 (2011).

[76] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 2000. The earth mover's distance as a metric for image retrieval. *International journal of computer vision* 40, 2 (2000), 99–121.

[77] Stan Salvador and Philip Chan. 2007. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis* 11, 5 (2007), 561–580.

[78] Peggy Seriès and Aaron Seitz. 2013. Learning what to expect (in visual perception). *Front. Hum. neurosci.* 7 (2013).

[79] Evan Shelhamer, Jonathan Long, and Trevor Darrell. 2017. Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 4 (2017).

[80] Chengyao Shen and Qi Zhao. 2014. Webpage saliency. In *European conference on computer vision*. Springer, 33–46.

[81] Jeremiah D. Still and Christopher M. Masciocchi. 2010. A Saliency Model Predicts Fixations in Web Interfaces. In *Proc. MDDAUI Workshop*.

[82] Michael J Swain and Dana H Ballard. 1991. Color indexing. *International journal of computer vision* 7, 1 (1991), 11–32.

[83] Hamed R. Tavakoli, Fawad Ahmed, Ali Borji, and Jorma Laaksonen. 2017. Saliency Revisited: Analysis of Mouse Movements Versus Fixations. In *Proc. CVPR*.

[84] Sauer Tim, A Yorke James, and Casdagli Martin. 1991. Embedology. *Journal of statistical Physics* 65, 3-4 (1991), 579–616.

[85] Richard Veale, Ziad M. Hafed, and Masatoshi Yoshida. 2017. How is visual salience computed in the brain? Insights from behaviour, neurobiology and modelling. *Philos. Trans. R. Soc. Lond. B. Biol. Sci.* 372, 1714 (2017).

[86] Ashish Verma and Debashis Sen. 2019. HMM-based Convolutional LSTM for Visual Scanpath Prediction. In *2019 27th European Signal Processing Conference (EUSIPCO)*. 1–5. https://doi.org/10.23919/EUSIPCO.2019.8902643

[87] Eleonora Vig, Michael Dorr, and David Cox. 2014. Large-Scale Optimization of Hierarchical Features for Saliency Prediction in Natural Images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[88] Wei Wang, Cheng Chen, Yizhou Wang, Tingting Jiang, Fang Fang, and Yuan Yao. 2011. Simulating human saccadic scanpaths on natural images. In *CVPR 2011*. IEEE, 441–448.

[89] Yixiu Wang, Bin Wang, Xiaofeng Wu, and Liming Zhang. 2017. Scanpath estimation based on foveated image saliency. *Cognitive processing* 18, 1 (2017).

[90] Alexa Top 500 Websites. 2022. https://www.expireddomains.net/alexa-top-websites/.

[91] Calden Wloka, Iuliia Kotseruba, and John K. Tsotsos. 2018. Active Fixation Control to Predict Saccade Sequences. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. https://doi.org/10.1109/cvpr.2018.00336

[92] Jeremy M. Wolfe and Todd S. Horowitz. 2004. What attributes guide the deployment of visual attention and how do they do it? *Nat. Rev. Neurosci.* 5, 6 (2004).

[93] Jason Wu, Xiaoyi Zhang, Jeff Nichols, and Jeffrey P Bigham. 2021. Screen Parsing: Towards Reverse Engineering of UI Models from Screenshots. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 470–483.

[94] Chen Xia, Junwei Han, Fei Qi, and Guangming Shi. 2019. Predicting Human Saccadic Scanpaths Based on Iterative Representation Learning. *IEEE Transactions on Image Processing* 28, 7 (2019), 3502–3515. https://doi.org/10.1109/TIP.2019.2897966

[95] Mulong Xie, Sidong Feng, Zhenchang Xing, Jieshan Chen, and Chunyang Chen. 2020. UIED: A Hybrid Tool for GUI Element Detection. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Virtual Event, USA) *(ESEC/FSE 2020)*. Association for Computing Machinery, New York, NY, USA, 1655–1659. https://doi.org/10.1145/3368089.3417940

[96] Pingmei Xu, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R Kulkarni, and Jianxiong Xiao. 2015. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755* (2015).

[97] Amir R. Zamir, Te-Lin Wu, Lin Sun, William B. Shen, Bertram E. Shi, Jitendra Malik, and Silvio Savarese. 2017. Feedback Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[98] Joseph P Zbilut and Charles L Webber Jr. 2006. Recurrence quantification analysis. *Wiley encyclopedia of biomedical engineering* (2006).

[99] Ciheng Zhang, Decky Aspandi, and Steffen Staab. 2022. Predicting Eye Gaze Location on Websites. *arXiv preprint arXiv:2211.08074* (2022).

[100] Qi Zhao and Christof Koch. 2013. Learning saliency-based visual attention: A review. *Signal Process.* 93 (2013).