SEEKUI: Predicting Visual Search Behavior on Graphical User Interfaces with a Reward-Augmented Vision Language Model

Zixin Guo*
zixin.guo@aalto.fi
Aalto University
Finland

Yue Jiang*
yue.jiang@utah.edu
Aalto University
Finland
University of Utah
United States

Luis A. Leiva
name.surname@uni.lu
University of Luxembourg
Luxembourg

Antti Oulasvirta
antti.oulasvirta@aalto.fi
Aalto University & ELLIS Institute Finland
Finland

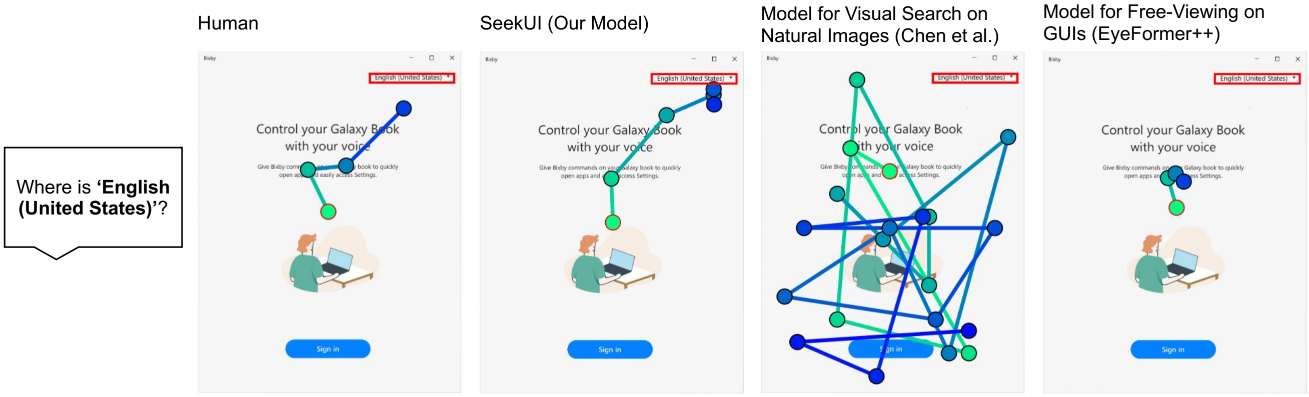

Where is 'English (United States)'?
Human
SeekUI (Our Model)
Model for Visual Search on Natural Images (Chen et al.)
Model for Free-Viewing on GUIs (EyeFormer++)
Bixby
English (United States)
Control your Galaxy Book with your voice
Give Bixby commands on your Galaxy book to quickly open apps and easily access Settings.
Sign in

Figure 1: Given a GUI screenshot and a text cue of a target element (in this example, "English (United States)"), SEEKUI predicts a scanpath of how users would search for such an element on the GUI. Scanpaths are here visualized with a color gradient (green → blue) indicating temporal progression, with fixation points marked as circles. The target element is highlighted in a red bounding box. The ground-truth human scanpath is closely approximated by our model, which reproduces human-like search strategies. In contrast, the state-of-the-art model by Chen et al. (a natural-image visual search model) and EyeFormer (a free-viewing GUI model) fail to replicate the pattern.

Abstract

Visual search is key to understanding and improving interaction with graphical user interfaces (GUIs), yet predicting scanpaths on real GUIs remains an open challenge. Unlike free-viewing, visual search is goal-driven and shaped by both linguistic and visual features of the GUI. State-of-the-art models of visual search, trained on natural images, fail with GUIs because they cannot capture the effects of grouping and semantics on search strategies. We present SEEKUI, a reward-augmented Vision Language Model (VLM) that predicts scanpaths directly from a GUI screenshot and a text cue describing the desired target. Our model extends the capability of VLMs to reproduce human-like visual search behavior on GUIs and outperforms baseline models across different types of GUIs. Importantly, it reproduces key empirical phenomena established in eye-tracking studies of visual search, including the Guess–Scan–Confirm strategy. In sum, SEEKUI provides a foundation for predicting visual search behavior and has potential for informing GUI evaluation and optimization.

*Both authors contributed equally to this research.

CHI '26, Barcelona, Spain
© 2026 ACM.
ACM ISBN 979-8-4007-2278-3/2026/04
https://doi.org/10.1145/3772318.3791178

CCS Concepts

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

Keywords

Eye Tracking, User Interface, Vision Language Model, Visual Search, Reinforcement Learning

ACM Reference Format:

Zixin Guo, Yue Jiang, Luis A. Leiva, and Antti Oulasvirta. 2026. SeekUI: Predicting Visual Search Behavior on Graphical User Interfaces with a Reward-Augmented Vision Language Model. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 23 pages. https://doi.org/10.1145/3772318.3791178

1 Introduction

Visual search, the task of locating a specific object among distractors, is a fundamental task in everyday interaction with Graphical User Interfaces (GUIs) [39, 84]. Every time users look for an icon in a toolbar, a menu item in a list, or a button on a form, they engage in visual search. This process occurs countless times in daily computer and mobile use, across applications and platforms. When users fail to find the desired object quickly, tasks slow down, frustration mounts, and overall usability suffers [66, 69, 101]. Accurately modeling these patterns is therefore essential for designing interfaces that are both effective and satisfying for users. A promising approach is to develop computational models to predict *scanpaths* on GUIs; i.e., the sequence of fixations that users make while exploring the interface.

In predicting scanpaths, it is challenging to capture the sequence of gaze shifts, not merely their static locations. Spatial density estimates, like heatmaps, tell where people look, but they overlook the critical temporal order of fixations. This sequential perspective is important, as the order of fixations reveals a user's underlying visual search strategy. Here, on GUIs, visual search is not a passive observation of arbitrary visual features but an active navigation through a graphical structure designed to guide attention. This fundamentally distinguishes GUI visual search from searching in natural images [13, 76] or during free-viewing [23, 44–46, 107]. Unlike in natural scenes, GUI search is strongly guided by the interface's designed structure, such as its spatial organization, grouped elements (e.g., menus, toolbars), and semantic relationships [13, 76]. Its goal-directed nature sets it apart from exploratory free-viewing; search requires users to integrate a target cue with their knowledge of the layout [84]. Consequently, scanpath prediction models developed for natural images and free-viewing are ill-suited for GUI visual search.

In this work, we present SeekUI, a scanpath prediction model for visual search on GUIs (Figure 1). SeekUI is designed to specifically address the dual gaps of layout understanding and guidance by a top-down task. Unlike some other models of search [33], SeekUI works directly from pixels. This is valuable for applications where there is no structured representation of the GUI available. The technical insight is to build on the capabilities of Vision Language Models (VLMs) for understanding both images and text. To better model GUI visual search, we design SeekUI to predict not only "where" a user will look (*scanpath modeling*), but also the "what and why" that motivates their gaze (*explanation modeling*). However, out of the box, VLMs are not able to replicate human-like scanpaths on GUIs. We therefore tune the VLM via reward-augmentation, guiding it to better match human strategies specifically in the way they exploit visual and semantic cues in GUI-based search [46, 61]. Technically, SeekUI builds upon the state-of-the-art VLM Qwen2.5-VL [4], which is pre-trained on a vast corpus of web data and provides a robust understanding of GUI components and layouts. Additionally, we implement an Instruction Tuning stage followed by a Reinforcement Learning (RL) stage for policy refinement.

To demonstrate SeekUI's capabilities, we compare it with two baseline categories: (1) free-viewing models for GUIs and (2) visual search models for natural images, adapting or fine-tuning all on a curated GUI visual search dataset [84]. SeekUI significantly outperforms these baselines on diverse GUIs, including mobile, desktop, and web interfaces. Its predicted gaze sequences replicate signature patterns of GUI visual search, including the empirically observed Guess-Scan–Confirm search strategy [57]: Early fixations are biased toward the top-left corner, reflecting learned priors about where relevant elements often appear on GUIs. This is followed by selective scanning that aligns with the GUI's spatial structure and the semantic features of the target. Finally, fixations converge on the target during the confirmation stage. The Guess–Scan–Confirm strategy is important because it reflects how users balance prior expectations, systematic exploration, and final verification, which is identified as a gaze pattern for goal-directed visual search on GUIs [57, 84]. In addition, SeekUI reproduces the characteristic distributions of saccade directions and lengths across interface types, achieves substantially higher success rates than baselines, and mirrors the empirical relationship between visual clutter and search time [84]. Thus, by capturing both the temporal sequence and spatial distribution of fixations, SeekUI not only produces more accurate predictions but also reproduces behavioral phenomena reported in prior empirical studies of GUI visual. Finally, ablation experiments confirm that the combination of explantion modeling and reinforcement learning improves the overall performance.

We introduce two application scenarios that leverage SeekUI's ability to simulate human-like search behavior: (1) *evaluation of individual element placement*, where designers can reposition interface components and instantly observe how the predicted scanpath changes; and (2) *full-layout optimization*, where multiple layout variants are compared using a quantitative search efficiency metric to identify designs that minimize predicted search time for important target elements. These applications illustrate how SeekUI can support rapid, early-stage prototyping without requiring eye-tracking data or user testing at each design iteration.

In summary, our work makes the following contributions:

- We present SeekUI, a VLM-based approach for predicting visual search scanpaths on GUIs, capable of handling mobile, desktop, and web interfaces.
- We show that an RL stage with a reward function defined over entire scanpaths enables SeekUI to recover from prediction errors and generate more accurate gaze sequences.

- We provide quantitative and qualitative evaluations on a curated GUI visual search dataset, demonstrating that SeekUI outperforms state-of-the-art baselines and accurately reproduces empirically observed visual search phenomena.
- We demonstrate two practical applications, element placement evaluation and full-layout optimization, showing how SeekUI can directly support GUI design workflows.

Open Science. We release our code, models, and data at https://github.com/YueJiang-nj/SeekUI-CHI2026.

2 Related Work

Our work builds on research in visual search models for GUIs, scanpath prediction, and VLMs. Below, we frame our approach within these areas.

2.1 Visual Search

Visual search has long been studied in psychology and HCI, but GUI search differs from simplified laboratory displays and natural scenes. Classic theories such as Feature Integration Theory [103] and Guided Search [109] highlight the limited capacity of human recognition, making selective attention essential. In simplified displays, search is guided by features like color, size, or orientation [103, 109, 110], whereas in natural scenes, semantic context and spatial structure also play a role [36, 37]. HCI research often draws on Information Foraging Theory [10, 83, 93], viewing search as navigation guided by information scent. GUIs differ, however, as they are highly structured environments with targets embedded in designed layouts such as menus, grids, and toolbars [43, 50, 52–54], requiring users to combine perceptual cues with learned expectations [35, 41, 42, 47–49, 71].

Putkonen et al. [84] provide the most comprehensive empirical account of GUI visual search, using eye-tracking across mobile, desktop, and web interfaces. They show that search is shaped by interface design and target semantics rather than low-level features, with consistent patterns in saccade directions (vertical in mobile, horizontal in desktop/web), heavy-tailed saccade lengths, and a strong dependence of search time on visual clutter rather than raw element count. They also identify the Guess–Scan–Confirm strategy, reflecting expectation-driven search, systematic exploration, and final verification.

Earlier HCI work noted related effects: spacing and grouping influence icon and menu search [5, 24], visual complexity impairs search [7], cultural conventions modulate scanning strategies [8], and adaptive strategies respond to layout density [104] or utilize assistive cues [22, 38]. Together, these studies highlight that GUI search depends on structure and expectations, not just perceptual saliency. SeekUI builds on these insights, capturing saccade distributions, success rates, clutter effects, and the Guess–Scan–Confirm strategy across GUI types.

Building on this foundation, our work connects directly to the phenomena identified by Putkonen et al. [84]. While prior research described these human behaviors, no computational model has reproduced them across GUI types [26, 51, 81, 97]. SeekUI fills this gap by capturing the same patterns in saccade distributions, success rates, clutter effects, and the Guess–Scan–Confirm strategy, providing a predictive model that aligns with both prior theory and empirical observation.

2.2 Modeling Visual Search on GUIs

Modeling visual search is particularly challenging on GUIs because they combine perceptual saliency with structured layouts and task-driven expectations. Prior studies show that search time increases with the number of elements, but actually clutter metrics and complexity measures better capture perceptual difficulty [74, 89]. Text targets are harder to locate than icons [116], and search performance is influenced by visual distinctiveness such as color contrasts or borders around app icons [69, 102]. These findings highlight that GUI search cannot be explained by bottom-up features alone, but requires models that integrate structural and semantic context [84].

Early computational models of GUI search examined specific tasks such as menu navigation, icon recall, or keyboard scanning across different interface components [5, 12, 32, 55]. While valuable for understanding task-specific strategies, these approaches were limited in scope and did not generalize across diverse GUI types.

Subsequent efforts sought to formalize GUI search through rule-based or crowdsourced approaches. Yuan and Li [116] used Deep Learning (DL) to predict search times on webpages, but their study relied on crowdsourced completion times, lacked eye-tracking data, and focused only on one interface type. Similarly, Halverson and Hornof [33] developed a model requiring manually coded interface representations (object positions, grouping, density) rather than raw GUI images, restricting its use to simple, text-heavy layouts.

Cognitive architectures introduced more principled accounts of user strategies in GUI search. CogTool-Explorer [99] integrated hierarchical search processes into IFT-based predictions and showed strong alignment with human performance across layouts. Other work emphasized active vision: Halverson and Hornof [34] proposed a minimal model grounded in cognitive strategies and oculomotor processes, while Kieras and Hornof [58] extended these ideas to account for eye movement dynamics and visual acuity. Such models highlight the role of higher-level cognitive factors but remain difficult to scale to naturalistic GUI data. However, they require specific representations for GUIs with detailed structure and element information as inputs, and thus cannot model visual search directly from pixels. In contrast, SeekUI combines a VLM with RL to predict full scanpaths directly from pixel-level GUI images. This design also allows it to model both spatial accuracy and temporal coherence, capturing the design-driven and task-driven search effects documented in empirical work [84].

2.3 Scanpath Prediction Models

Research on scanpath prediction has been dominated by free-viewing GUIs, where models attempt to generate gaze sequences without explicit search goals. Approaches can be broadly categorized into three directions: saliency-map generation, direct sequence modeling, and RL-based.

Saliency-map methods. Early work estimated scanpaths by sampling from saliency maps, sometimes combined with mechanisms such as inhibition-of-return (IOR) to discourage repeated fixations at the same location [40]. Wang et al. [105] pioneered the simulation of saccadic scanpaths under an information maximization framework by integrating sparse coding–based sensory responses with a decaying visual working memory, and selecting each fixation as the location of maximal residual perceptual information. Later studies refined these strategies by iteratively sampling fixations from saliency distributions or by adding handcrafted constraints [2, 17, 59, 72, 88, 106, 108, 112]. While effective at identifying regions of high visual interest, these methods often face certain constraints. For instance, they frequently prioritize spatial saliency over temporal factors such as fixation durations or consistent sequence ordering. Additionally, their typically non-differentiable nature can make seamless integration into end-to-end deep learning architectures more challenging.

Direct sequence modeling. To address the lack of temporal structure, subsequent work predicted fixation sequences explicitly. Some approaches generate points from parameterized Gaussian mixtures, as in IOR-ROI [17, 95] and Visual ScanPath Transformer [85]. Others have adopted newer sequence models, such as Transformer-based GazeFormer [76] or Markov-based ScanDMM [94]. Adversarial methods such as PathGAN and ScanGAN [3, 73] attempt to capture variability in human gaze but often produce unrealistic clustering or misplaced fixations. Although these models explicitly generate sequences, they are prone to error accumulation, where inaccuracies in early predictions propagate throughout the scanpath [87].

Reinforcement Learning approaches. Another line of research formulates scanpath prediction as a sequential decision-making problem. Early work explored RL with hand-crafted state representations to guide attention in cluttered scenes [75, 79]. More recent methods leverage deep RL, including inverse RL for visual search [114], deep RL for panoramic video scanpaths [113], and policy-gradient optimization with discretized fixation grids [13]. Grid-based discretization simplifies optimization but sacrifices spatial precision, motivating continuous-control formulations such as EyeFormer [44] and GazeXplain [14], which generate fixations from Gaussian policies with rewards optimizing the trajectories.

Together, previous approaches have advanced sequence prediction in free-viewing settings, but they are not designed for visual search on GUIs. SeekUI extends this line of work by combining VLMs with RL to generate scanpaths conditioned on both visual layout and explicit task cues, enabling prediction of task-driven search behavior beyond free-viewing paradigms.

2.4 Vision Language Models

VLMs integrate visual and textual inputs into joint representations that support a wide range of multimodal tasks, including captioning [29, 30, 63, 115], retrieval [31, 64, 65], and instruction following [1, 15, 20, 68]. By grounding visual perception in natural language, these models enable conditioning on explicit task descriptions, making them particularly suitable for situations where users search for targets specified by text.

Recent VLMs have demonstrated rapid progress in both general-purpose and domain-specific applications. For example, LLaVA [68] introduced a visual instruction tuning framework that combines CLIP's powerful vision encoder [86] with Vicuna [18], a large language model (LLM) trained on instruction-following data. Building on this foundation, LLaVA-Next [67] incorporated architectural and training improvements, closing the gap with state-of-the-art commercial systems such as Google Gemini. These systems illustrate how multimodal instruction tuning can produce flexible, instruction-aligned VLMs.

Several VLMs have been recently designed specifically for GUIs. Spotlight [62], for example, takes a GUI screenshot and a region of interest as input, and outputs relevant text for tasks such as widget captioning, screen summarization, or tappability prediction. ILuvUI [51] demonstrated how a VLM-based agent can handle broader GUI tasks that require reasoning and multimodal grounding, such as answering visual questions within the GUI context. These systems highlight that GUI-specific VLMs can learn structured representations of interface elements, extending beyond the natural image domain. However, while VLMs excel at visual-text understanding and natural language generation, they are not inherently designed to model sequential gaze behavior or perform visual search tasks. Standard VLM outputs are static and do not capture the temporal dependencies between fixations, nor do they account for task-driven search strategies on GUIs.

SeekUI builds on these insights by extending a pretrained state-of-the-art VLM (Qwen, which includes GUI-related data in its training corpus) to generate task-driven scanpaths on GUIs. By conditioning on both GUI appearance and the text cue on the target, we guide the VLM to understand the visual search task and to produce scanpaths that reflect both spatial structure and temporal dynamics. We further refine the model by optimizing for scanpath-level similarity using reinforcement learning, enabling more accurate modeling of human-like sequential gaze behavior.

3 Method

Existing scanpath prediction models for natural-image free viewing are ill-suited for GUI visual search, which requires both understanding interface structure and integrating the user's task. To address these challenges and inspired by GazeXplain [14], SeekUI predicts not only *where* a user will look, but also *what and why* drives each fixation. This design aligns with the *two-stream hypothesis* [27]: the ventral stream identifies objects ("what"), while the dorsal stream guides actions ("where/how"). Accordingly, SeekUI formulates visual search as a text generation process with two components: (1) explanation modeling, which outlines the search strategy, and (2) scanpath modeling, which predicts the fixation sequence.

SeekUI is trained in two stages. First, instruction tuning teaches the model to generate explanation–scanpath pairs from human ground truth data. Second, RL fine-tuning encourages the production of coherent, human-like scanpaths using a non-differentiable sequence-level reward.

3.1 Visual Search Formulation

To tackle GUI structure understanding and task integration, we explicitly generate a strategic explanation describing how a user would search for the target. This approach solves both challenges:

- Structure understanding: The explanation identifies and reasons about the functional and spatial layout of relevant GUI regions.

- Task integration: The explanation connects the regions to the user's goal until the target is found.

Thus, SeekUI first models *why* a gaze shift occurs and then predicts *where* the gaze moves. Formally, given a GUI screenshot I and a textual target cue T, the model needs to generate a textual sequence containing two components: a natural language explanation $\mathbf{E}$, followed by the scanpath $\mathbf{F} = (f_1, f_2, \ldots, f_n)$. Each fixation point $f_i = (x_i, y_i)$ represents a spatial location on the interface. Our objective is to learn a function, SeekUI$(\cdot, \cdot)$, that maps both inputs to a unified output:

$$\text{SeekUI}(I, T) = [\mathbf{E}; \mathbf{F}]. \tag{1}$$

where $[\cdot\ ;\ \cdot]$ denotes sequence concatenation.

3.2 Model

GUI visual search demands a model with the dual capabilities of perceiving fine-grained visual details (such as icons and text) and aligning the visual context with a textual target cue. To meet these requirements, we build SeekUI upon a VLM architecture, for its powerful capabilities in: (1) pre-trained knowledge of GUI structures, and (2) multimodal processing of GUI screenshots and text cues.

Model Architecture. The VLM architecture acts to comprehend and reason about visual inputs. It is composed of three main components: (1) a Vision Transformer (ViT) based vision encoder, which ingests the input GUI screenshot and encodes it into a sequence of spatially-aware patch embeddings; (2) an MLP-based Projector acting as a bridge, mapping features in the visual embedding space into the LLM space; and (3) an LLM serving as the reasoning and generative core, conditioning its generation on the aligned visual features. Architecturally, this design leverages the vision encoder for high-fidelity spatial contexts, the projector for cross-modal fusion, and the powerful LLM to perform complex, visually-conditioned generation and reasoning tasks.

Model Input. The VLM takes as input a GUI screenshot along with the corresponding textual instruction that contains the target cue. To ensure that the VLM learns to produce scanpath coordinates within the correct spatial range, the resolution of the GUI screenshot is also included in the textual instruction. This textual instruction is constructed as:

> *Given the image with width {width} and height {height}, what is the scanpath for the visual search task on this GUI? The text on the target element is "{target}".*
> *Output the explanation process in <explanation> </explanation> and final answer in <answer> </answer> tags. The output answer format should be as follows:*
> *<explanation> ... </explanation> <answer>The scanpath is [x1, y1] [x2, y2] ...</answer>*
> *Please strictly follow the format.*

Here, "{width}" and "{height}" are the placeholders for the width and height of the GUI screenshot. By explicitly including the resolution (width, height) in the prompt, we ground the model's coordinate prediction mechanism, allowing it to normalize spatial features relative to the canvas size.

Model Output. We structure the ground truth of VLM's output using two specialized token pairs. The first pair, <explanation> and </explanation>, is responsible for the explanation modeling, guiding the model to generate search strategy. The second pair, <answer> and </answer>, handles the scanpath modeling, directing the model to output the corresponding fixation sequence. This structure forces the model to first explain verbally a rationale and then generate the corresponding fixation sequence, promoting consistency in the generated scanpath. The target output is a single ground truth sequence, *e.g.,* "<explanation> The user starts by scanning the top navigation bar... </explanation> <answer> The scanpath is [540, 100] [540, 300]... </answer>".

Formally, a structured, concatenated ground-truth string $\mathbf{S}$ is constructed on explanation $\mathbf{E}$ and fixation $\mathbf{F}$ data as:

$$\text{Serialize}(\mathbf{F}) = \text{"The scanpath is } (x_1, y_1)\ (x_2, y_2)\ \ldots\text{"}, \tag{2}$$

$$\begin{aligned}\mathbf{S} = &\ \text{<explanation>}\mathbf{E}\text{</explanation>}\\ &\ \text{<answer>Serialize}(\mathbf{F})\text{</answer>},\end{aligned} \tag{3}$$

where Serialize$(\cdot)$ is the function that converts the fixation coordinate sequence $\mathbf{F}$ into text. This string $\mathbf{S}$ is then tokenized as the final ground-truth token sequence $\hat{S}$ that the VLM needs to predict:

$$\hat{S} = \text{Tokenize}(\mathbf{S}). \tag{4}$$

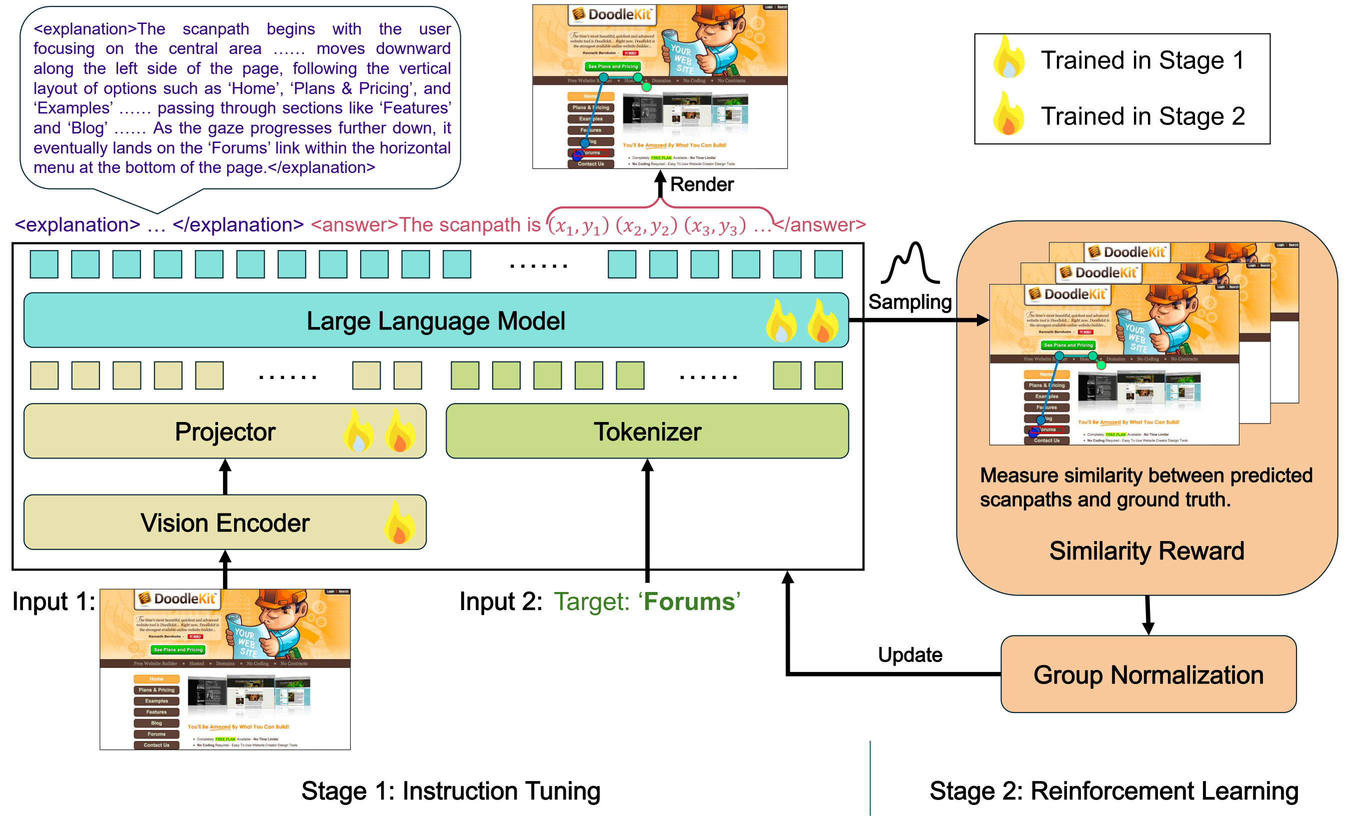
Directly prompting the VLM to generate a scanpath often results in incoherent, non-human-like outputs. The main reason is that a pre-trained VLM, while powerful, is fundamentally designed for generic understanding tasks like captioning or question answering. When performing visual search, it lacks the intrinsic mechanisms to bridge the significant semantic gap between identifying a textual goal and performing the action-oriented scanpath sequence. To close this semantic gap, we employ a two-stage training for SeekUI: an instruction tuning stage, followed by an RL stage that refines the model's ability to generate coherent, human-like scanpaths.

3.3 Stage (1): Instruction Tuning

To bridge the semantic gap between static GUI screenshots and dynamic scanpath generation, we employ a supervised instruction tuning stage. This phase serves two purposes: it aligns the VLM's visual encoder with the specific domain of scanpath prediction, and it teaches the model the structural format of our required output (explanation followed by coordinate sequence).

3.3.1 Data Preparation and Explanation Synthesis. We fine-tune SeekUI for visual search on a curated subset of the VSGUI10K dataset [84], which provides GUI screenshots, textual target cues, and corresponding human scanpaths. However, raw scanpath coordinates (x, y) lack the semantic reasoning required for the model to "understand" the search strategy. To address this, we augment the dataset with natural language explanations.

Since ground-truth explanations are not available in VSGUI10K, we distill knowledge from a VLM model, Qwen2.5-VL-72B-Instruct [4], to automatically generate them. To ensure the model accurately interprets the spatial data, we employ a "visual prompting" strategy. We directly visualize the human scanpath on the GUI image by plotting the fixation sequence: marking the starting point with a green circle (red boundary) and the ending point with a blue circle (black boundary), as shown in Figure 1. This visual overlay



<explanation>The scanpath begins with the user focusing on the central area …… moves downward along the left side of the page, following the vertical layout of options such as 'Home', 'Plans & Pricing', and 'Examples' …… passing through sections like 'Features' and 'Blog' …… As the gaze progresses further down, it eventually lands on the 'Forums' link within the horizontal menu at the bottom of the page.</explanation>

<explanation> … </explanation> <answer>The scanpath is (x_1, y_1) (x_2, y_2) (x_3, y_3) …</answer>
Render
Trained in Stage 1
Trained in Stage 2
Sampling
Large Language Model
Projector
Tokenizer
Vision Encoder
Input 1:
Input 2: Target: 'Forums'
Measure similarity between predicted scanpaths and ground truth.
Similarity Reward
Update
Group Normalization
Stage 1: Instruction Tuning
Stage 2: Reinforcement Learning

Figure 2: Overview of SeekUI. Given a GUI screenshot and a target text cue, SeekUI predicts a human-like scanpath for locating the target. Stage (1): Instruction Tuning with explanations. The vision encoder and projector align GUI features with an LLM, which generates explanations and scanpaths conditioned on the target. Stage (2): Reinforcement Learning with a similarity reward. Predicted scanpaths are compared with human data, and the reward signal updates the model via group normalization. This two-stage design allows SeekUI to capture both spatial-semantic grounding and temporal search dynamics.

removes the ambiguity of describing complex coordinates via text. We then prompt the model with the scanpath-visualized image and the following instruction to generate the reasoning:

> *Explain the explanation process behind the visual scanpath for the target "{target}". The scanpath begins at the green circle with the red boundary, which is usually located at the center or appear in other regions. Write one paragraph describing the entire explanation process, ensuring the paragraph covers all areas where fixation points occur and following the order of fixation points to explain. Don't describe areas where fixation points are not located. Here are two examples of how to explain this for other GUIs and scanpaths. Do not mention any green/blue circles or lines. Use English!*

"{target}" is a placeholder for the respective target textual cue. We present two examples of the generated ground truth explanation:

> *Example 1: The scanpath begins with the user fixating on the Magic Quadrant chart, likely because it is visually distinct and initially captures attention, before moving left to the central promotional text and "Read the Report" button to check for relevant actions. Not finding the sign-up option there, the user shifts gaze upward to the top navigation bar, following a common expectation that account-related actions are located at the top right. They first scan nearby links like "Join a Meeting" and "Host a Meeting," then continue rightward until finally identifying the "Sign Up, It's Free" button, successfully completing the search.*

> *Example 2: The scanpath starts with the user looking at the center of the screen near the "Online Flash Event" banner, then moving upward to the top-left logo and promotional text, likely checking for navigation cues or sale-related links. The gaze then shifts rightward toward the model image, perhaps drawn by its visual prominence, before moving sharply downward toward the lower navigation buttons. After scanning these icons, the user's attention is drawn to the bright red "CLEARANCE" button on the lower right, where they fixate to confirm it matches the target. This sequence suggests an initial top-down search in expected navigation areas, followed by exploration of visually salient content, and finally a bottom-up attention shift to the highly contrasted target button.*

The resulting dataset consists of triplets: $(I, T, E + F)$, where I is the clean GUI screenshot, T is the target cue, and $E + F$ is the synthesized explanation combined with the ground truth coordinate sequence.

3.3.2 Optimization Objective. The model is optimized in a supervised manner using a standard cross-entropy (CE) loss. The visual search is formulated as an autoregressive text generation process, where the model predicts the next token in the sequence (both explanation words and coordinate numbers) based on the image embeddings and previous tokens.

Formally, the probability of generating the sequence $\hat{S}$ (comprising both the explanation E and fixations F) is factorized autoregressively according to the chain rule of probability:

$$p_\theta(\hat{S}|I, T) = \prod_{t=1}^{l} p_\theta(\hat{s}_t|\hat{s}_{<t}, I, T), \tag{5}$$

where l denotes the sequence length of $\hat{S}$, which is the combined explanation and scanpath, $\hat{s}_{<t}$ denotes the sequence of preceding ground-truth tokens $(\hat{s}_1, ..., \hat{s}_{t-1})$, and θ represents the parameters of SeekUI. When maximizing the conditional log-likelihood of the ground-truth sequence across the entire training dataset, it is equivalent to minimizing the standard CE loss:

$$\mathcal{L}_{\text{CE}} = -\sum_{t=1}^{l} \log p_\theta(\hat{s}_t|\hat{s}_{<t}, I, T). \tag{6}$$

This optimization enables the model to capture characteristic human gaze patterns, including the dynamic scanning strategies, systematic exploration, and revisits to salient elements.

However, the instruction tuning stage forces the model to strictly mimic the ground-truth sequence. The model learns to predict the next token based on a perfect, preceding ground-truth history, creating a discrepancy with real-world inference where the model must rely on its own, potentially imperfect, predictions. This mismatch leads to accumulated errors: a single slight deviation can cause the next prediction to be further off, leading to a sequence that rapidly drifts and fails to converge on the target.

3.4 Stage (2): Reinforcement Learning

Since the model generates autoregressively, each prediction depends entirely on the preceding sequence. During the instruction tuning process, the model is guided by the ground truth history, effectively masking any prior errors. However, during inference, the model must rely on its own past predictions, meaning a single deviation can cause the entire sequence to drift. To build robustness against these imperfect predictions, we employ a stage where the model learns directly from its self-generated sequences rather than ground truth. In addition, unlike the instruction tuning stage, which optimizes for per-token accuracy, this stage focuses on the holistic quality of the generated trajectory. To this end, SeekUI further employs a RL stage to directly optimize for globally coherent scanpath sequences.

3.4.1 Environment, Policy, State, and Action. During RL fine-tuning, the VLM operates as a decision-making agent. The *environment* is the static context provided by the GUI screenshot and the target cue. The agent's *policy* is the VLM, denoted as π_θ. At each step, the agent observes the current *state*, which includes the static environment and the complete history of previously generated explanations and fixations. Based on the state, the agent's *action* is to generate a structured sequence composed of the explanation and the scanpath. Once a full scanpath is generated, a *reward* is then provided to the agent, calculated from the predicted sequence of scanpath coordinates and the corresponding ground truth.

3.4.2 Reward Function. To measure the spatial accuracy and temporal order of fixations, we incorporate a ScanMatch [13, 19] reward. Standard spatial metrics (e.g., Mean Squared Error or Intersection over Union) treat fixations as independent points, ignoring the sequential dependencies critical to search strategies. In contrast, ScanMatch measures the similarity between predicted and human scanpaths at the sequence level, capturing both the spatial alignment of fixations and their temporal order. Let the predicted scanpath be $\hat{F} = (\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_n)$ and the human scanpath be $H = (h_1, h_2, \ldots, h_m)$. The ScanMatch score is computed using the Needleman-Wunsch algorithm [77], which aligns two sequences by maximizing their overall similarity based on edit distance (substitution, insertion, and deletion operations):

$$\text{Dist}(i, j) = \max \begin{cases} \text{Dist}(i-1, j-1) + \text{Sub}(\hat{f}_i, h_j), & \text{(match/substitute)} \\ \text{Dist}(i-1, j) - g, & \text{(deletion)} \\ \text{Dist}(i, j-1) - g, & \text{(insertion)} \end{cases} \tag{7}$$

where $\text{Sub}(\hat{f}_i, h_j)$ is the substitution score reflecting the similarity between fixations, typically based on spatial distance, and g is a gap penalty for insertions or deletions ($g = 1$ in our implementation). The alignment matrix $\text{Dist}(i, j)$ is initialized as:

$$\text{Dist}(0, 0) = 0, \quad \text{Dist}(i, 0) = -i \cdot g, \quad \text{Dist}(0, j) = -j \cdot g. \tag{8}$$

The final ScanMatch reward is normalized to fall within the $[0, 1]$ range:

$$R_{\text{ScanMatch}}(\hat{F}, H) = \frac{\text{Dist}(n, m)}{\max(n, m)}, \tag{9}$$

where a score of 1 indicates identical scanpaths and 0 indicates no similarity.

3.4.3 Optimization. To perform a scalable learning strategy while avoiding the computational overhead of policy evaluation using a critic model [91], we employ Reinforcement Learning with Verifiable Rewards (RLVR) [92, 98] for this stage, specifically leveraging the Group Relative Policy Optimization (GRPO) algorithm. Unlike traditional approaches such as Proximal Policy Optimization (PPO) which rely on a separate value function, GRPO optimizes the current policy by sampling a group of G candidate responses for each input and leveraging their relative rewards (advantages) to guide policy updates. This approach is particularly effective for reasoning tasks where the ground truth (scanpath) allows for deterministic verification.

The model optimizes the current policy π_θ by sampling G candidate responses for each input and leveraging their relative rewards to guide policy updates. To reduce the variance of the gradient estimate and stabilize the training process, we compute the *group normalization* on the rewards. Specifically, given a sampled group of responses $\hat{S}^o = [\hat{S}_1, \ldots, \hat{S}_G]$, we first extract the corresponding responses of scanpath $\hat{F}^o = [\hat{F}_1, \dots, \hat{F}_G]$. Then we compute their normalized rewards (aka *advantages*) $A^o = [A_1, \dots, A_G]$ as:

$$r_i = \begin{cases} R_{\text{ScanMatch}}(\hat{F}_i, H), & \text{if } \hat{F}_i \text{ is extracted successfully from } \hat{S}_i, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

$$A_i = \frac{r_i - \mu(\{r_1, \dots, r_G\})}{\sigma(\{r_1, \dots, r_G\})}, \tag{11}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ represent the mean and standard deviation of the rewards within the sampled group, respectively. The training objective maximizes the likelihood of better-performing responses while penalizing deviations from the reference model π_θ^{ref}:

$$\mathcal{L}_{\text{RL}} = -\mathbb{E}_{\hat{S}^o \sim \pi_\theta(I,T)} [A^o] + \beta \cdot \text{KL}(\pi_\theta(\hat{S}^o \mid I, T) \parallel \pi_\theta^{\text{ref}}(\hat{S}^o \mid I, T)), \tag{12}$$

where β is the hyperparameter to control the KL-divergence between policies. The implementation details of our SeekUI are provided in the Supplementary Materials.

4 EVALUATION METHOD

We evaluate SeekUI through qualitative comparisons, quantitative evaluations, and analysis of human characteristics in search to assess its ability to generate human-like visual search scanpaths on GUIs. Our experiments address two main research questions:

(1) How does SeekUI compare with baseline models in producing realistic scanpaths?
(2) To what extent does SeekUI replicate human search efficiency and behavioral patterns?

Since no prior model is designed to directly generate scanpaths for visual search on GUIs given an image and a text target cue, we adapt two categories of state-of-the-art model baselines for fair comparison. (1) models developed for free-viewing for GUIs [44], and (2) models designed for visual search in natural images [13, 76]. All baseline models are retrained or fine-tuned on our curated GUI visual search dataset (see next section) to ensure a fair cross-system comparison. Together, these baselines allow us to test whether models optimized for free-viewing or natural scenes can generalize to GUI search, and whether SeekUI captures the human strategies they fail to reproduce.

4.1 Dataset

The VSGUI10K dataset [84] is a public resource that was introduced as a benchmark for evaluating the *human-likeness* of computational models of visual search. The dataset, collected using a GazePoint GP3 eye tracker, includes scanpaths for GUIs (mobile, desktop, and web) and target element representations based on images and texts. For our experiments, we only considered the trials that have text cues, resulting in 730 GUI screenshots for analysis, comprising 1,010 specific target elements and 1,616 corresponding human scanpaths recorded from N=84 participants (42 male, 40 female, 2 non-binary, 71 were aged 18–30, nine 31–50, and four over 50).

For model training and evaluation, we adopt a randomized 85/15 split: 85% of the data (1,366 scanpaths from 850 image-target pairs) are used for training, reserving a small set of 273 scanpaths for model validation at every epoch, and 15% of the data (250 scanpaths from 158 image-target pairs) are used for testing.

Overall, the VSGUI10K dataset provides rich multimodal information linking GUI visual structure, target text cues, and sequential gaze behavior. It allows SeekUI to learn the relationships between interface layouts, text cues, and human scanpaths, supporting the generation of fixation sequences for visual search across diverse real-world GUIs.

4.2 Baseline Models

We compare SeekUI against three representative state-of-the-art baselines, spanning both GUI free-viewing and natural-image search models:

- EyeFormer [44]: A Transformer-based model for predicting gaze sequences on GUIs under free-viewing. To adapt it for visual search, we replace its vision encoder with RoBERTa-extracted [70] textual embeddings, yielding a variant we call EyeFormer++.
- GazeFormer [76]: A Transformer-based model designed for visual search in natural images, employing spatiotemporal attention to capture sequential dependencies in scanpaths.
- Chen et al. [13]: A CNN–RNN model for natural-image visual search, which achieves the state-of-the-art scanpath prediction results on natural images through supervised sequence modeling.

All baselines are retrained or fine-tuned on our curated dataset to ensure a fair comparison with SeekUI.

4.3 Metrics

We evaluate scanpath prediction using a comprehensive set of metrics that assess sequence alignment, spatio-temporal dynamics, and distributional consistency.

- **ScanMatch.** ScanMatch [19] encodes fixations by dividing the GUI into a grid and aligns the resulting symbolic sequences using the Needleman–Wunsch algorithm. A distance-based substitution matrix incorporates spatial relationships. Scores are normalized to $[0, 1]$, with higher values indicating better spatial and temporal alignment.
- **String-Edit Distance (SED).** SED [9, 25] computes the minimum insertions, deletions, and substitutions needed to transform one fixation string into another, assigning uniform cost to each operation. Lower SED indicates fewer corrections required to match human behavior.
- **Sequence Score (SS).** SS [114] converts scanpaths into strings of fixation cluster IDs and uses a string-matching algorithm [77] to quantify similarity. String matching then quantifies similarity, allowing more flexible alignment than raw coordinates.
- **MultiMatch.** MultiMatch [21] compares scanpaths along five dimensions: shape, direction, length, position, and duration by reducing them to saccade-vector sequences and aligning them to minimize combined differences. Each dimension yields a normalized score in $[0, 1]$, enabling fine-grained behavioral analysis.
- **Scaled Time-Delay Embedding (STDE).** STDE [105] embeds sliding-window subsequences into a multidimensional space and computes the average minimum Euclidean distance across embeddings, capturing spatial proximity and temporal ordering. Higher values reflect greater spatio-temporal similarity.
- **Pearson's Correlation Coefficient (CC).** CC [60] measures linear correlation between predicted and ground-truth saliency

maps, treating pixel intensities as random variables and normalizing covariance by their standard deviations. Scores range from −1 to 1, with higher values indicating stronger correspondence.

- **Area Under ROC Curve (AUC).** AUC [11, 56] evaluates fixation prediction as a binary classification problem by computing true and false positive rates across thresholds. It reflects the probability that a fixation is assigned higher saliency than a non-fixation.
- **Shuffled AUC (sAUC).** sAUC [56] is a variant of AUC that samples negatives from fixation locations in other images, mitigating center bias and producing a more task-driven evaluation. It penalizes generic center biases and rewards task-specific spatial predictions.
- **Normalized Scanpath Saliency (NSS).** NSS [82] measures the mean normalized score at human fixation locations. The predicted scanpath is converted to a Gaussian-blurred map (with σ approximating foveal radius), normalized to zero mean and unit variance, and evaluated at ground-truth fixation coordinates. NSS=0 indicates chance; higher values reflect accurate fixation localization and impose stronger penalties on false positives than AUC.

Together, these metrics capture complementary aspects of similarity: sequence-based alignment (ScanMatch, SED, SS), spatio-temporal trajectory structure (MultiMatch, STDE), and distributional agreement (CC, AUC, sAUC, NSS).

5 Results

We evaluate SeekUI through a comprehensive set of illustrative cases and quantitative analyses designed to assess both accuracy and human-like behavior. Our evaluation spans diverse GUI types and interface complexities. We first present comparisons between predicted scanpaths, ground truth, and state-of-the-art baselines—visual-search models for natural images (GazeFormer and Chen et al.) and a free-viewing GUI model adapted for search (EyeFormer++). Next, we report quantitative results using standard scanpath prediction metrics. We then examine whether SeekUI reproduces documented human search characteristics [57, 84], including success rates, saccade patterns, and the Guess–Scan–Confirm strategy. Finally, we conduct ablations to isolate the contribution of individual model components.

Overall, SeekUI: (1) predicts more accurate scanpaths than baseline models, both qualitatively and quantitatively; and (2) better reproduces human-like search characteristics, including success rates, saccade statistics, and high-level search strategies.

5.1 Illustrative Cases

Figure 3 shows five randomly selected examples across GUI types where users search for target elements (highlighted in red). Additional comparisons are provided in the Supplementary Materials.

EyeFormer++ scatters fixations across the interface, producing dense but task-irrelevant clusters, reflecting its optimization for exploratory free-viewing rather than goal-directed search. Natural-image search models struggle with GUI structure and text cues: GazeFormer often fails to locate the target region, with fixations drifting toward visually salient but irrelevant areas, while Chen et al.'s model distributes fixations broadly across the entire screen, leading to inefficient search. Furthermore, the out-of-box VLM generates scanpaths that either traverse sequentially from top-left to bottom-right or simply cluster at the center. These rigid patterns demonstrate a failure to understand visual search dynamics, due to a lack of alignment with human behaviors.

In contrast, SeekUI captures both the overall trajectory and the spatial distribution of fixations more accurately. By incorporating text cues and layout understanding, it reliably identifies relevant interface elements and reproduces temporal and spatial patterns characteristic of human scanpaths. These illustrative cases indicate that SeekUI better reflects human strategies in goal-directed GUI search than both GUI free-viewing and natural-image search baselines.

5.2 Accuracy of Scanpath Prediction

We evaluate scanpath prediction using the full suite of metrics described in Section 4.3, covering sequence alignment (ScanMatch, SED, SS), spatio–temporal trajectory properties (MultiMatch, STDE), and distributional consistency (CC, AUC, sAUC, NSS). Together, these metrics provide a comprehensive measure of correspondence between predicted and human visual behavior.

As shown in Table 1, SeekUI achieves the strongest performance across nearly all metrics and GUI types. It improves upon the best baseline by 47% in ScanMatch and more than doubles the NSS score, reflecting better sequence alignment and fixation distribution. These gains are consistent across mobile GUIs, desktop GUIs, and webpages.

The only exceptions are the Vec and Len components of MultiMatch, where EyeFormer++ scores slightly higher. However, these values arise from clustered fixations that do not prioritize task-relevant targets and thus harm performance on other metrics (e.g., SS, NSS). SeekUI instead achieves balanced improvement across spatial and temporal dimensions, confirming its superior modeling of human-like scanpaths.

5.3 Reproducibility of Human Characteristics in Search

Scanpath metrics capture spatial and temporal similarity but not whether a model reproduces key behavioral properties of human visual search. GUI search is shaped by consistent patterns [57, 84]: imperfect success rates due to task difficulty, characteristic saccade distributions, sensitivity to visual clutter, and the structured Guess–Scan–Confirm strategy. We therefore evaluate whether SeekUI recovers these behaviors.

5.3.1 *Success Rates.*

The success rate, defined as the proportion of trials in which the target was successfully located, directly measures functional search performance. Unlike similarity metrics that assess scanpath resemblance to humans, success rate evaluates whether the predicted search actually finds the target.

In our dataset, humans achieve a 70% success rate, showing that even participants occasionally fail due to lapses in attention, ambiguous cues, or visually similar distractors. This benchmark underscores the inherent difficulty of GUI search tasks.

For model evaluation, a trial is considered successful if the last three fixations of the predicted scanpath fall within the foveal region of the target. With a viewing distance of 50–55 cm, the foveal radius

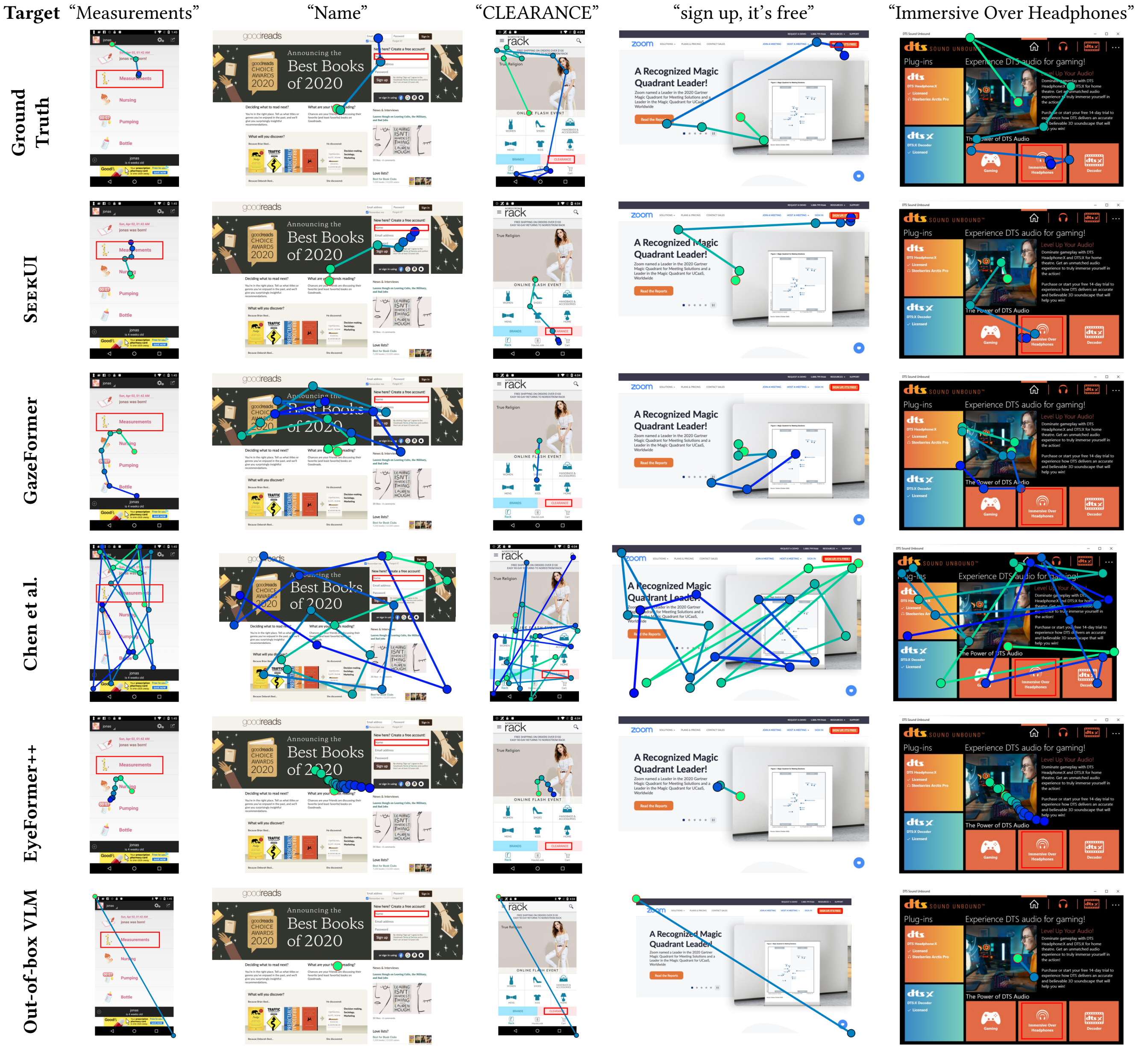


Figure 3: Qualitative comparison of scanpath predictions. Scanpaths are visualized with a color gradient (green → blue) indicating temporal progression, with fixation points marked as circles. Each column shows a GUI search task with the target element highlighted in a red bounding box, and each row shows scanpaths from ground-truth humans, SeekUI (ours), and baseline models. EyeFormer++ (free-viewing model) scatters fixations without prioritizing the task-relevant region. GazeFormer (natural-image search) often fails to locate the target, while Chen et al. (natural-image search) distributes fixations broadly across the interface. In contrast, SeekUI produces trajectories that more closely follow human patterns in both spatial distribution and temporal order compared to baselines, demonstrating its ability to generate realistic visual search behavior on GUIs.

is estimated at approximately 32 px. Landings within this radius indicate effective target acquisition.

Table 2 shows that SeekUI achieves 62% success, substantially higher than all baselines and close to human performance. GazeFormer and EyeFormer++ reach only 14% and 10%, respectively, while Chen et al.'s model achieves 18%. These low rates indicate that baseline models, despite plausible scanpath structures, fail to integrate task-driven cues, often wandering through distractors. In contrast, SeekUI not only reproduces human-like movement

Table 1: Accuracy of scanpath prediction across interface conditions (all, mobile, desktop, and web). Metrics are reported as mean values, with the best per column in bold. Arrows indicate the preferred direction (↑ = higher is better, ↓ = lower is better). SeekUI consistently outperforms baselines across nearly all metrics and GUI types, achieving stronger sequence alignment (ScanMatch, SS), lower dissimilarity (SED, STDE), and superior fixation distribution similarity (CC, AUC, sAUC, NSS). The only exceptions occur in the Vec and Len components of MultiMatch, where EyeFormer++ scores higher but reflects clustered fixations unrelated to task targets. Overall, the results demonstrate that SeekUI produces scanpaths that most closely mirror human search strategies on GUIs.

GUI Type	**Methods**	**ScanMatch ↑**	**MultiMatch**				**SED ↓**	**STDE ↑**	**SS ↑**	**CC ↑**	**AUC ↑**	**sAUC ↑**	**NSS ↑**
			Vec ↑	**Dir ↑**	**Len ↑**	**Pos ↑**							
All combined	GazeFormer	0.236	0.935	0.659	0.927	0.806	7.020	0.824	0.261	0.233	0.659	0.564	0.810
	Chen et al.	0.142	0.850	0.623	0.776	0.709	15.792	0.729	0.199	0.055	0.550	0.526	0.231
	EyeFormer++	0.181	**0.961**	0.608	**0.943**	0.797	10.596	0.833	0.165	0.203	0.625	0.546	0.677
	SeekUI	**0.347**	0.947	**0.677**	0.934	**0.855**	**5.776**	**0.860**	**0.320**	**0.410**	**0.747**	**0.659**	**1.3947**
Mobile	GazeFormer	0.242	0.931	0.668	0.923	0.816	5.861	0.829	0.292	0.242	0.663	0.551	0.813
	Chen et al.	0.139	0.848	0.617	0.785	0.716	15.393	0.734	0.206	0.061	0.543	0.512	0.221
	EyeFormer++	0.201	**0.951**	0.596	0.926	0.821	8.925	0.844	0.188	0.240	0.641	0.538	0.751
	SeekUI	**0.346**	0.944	**0.695**	**0.928**	**0.857**	**4.989**	**0.863**	**0.366**	**0.376**	**0.732**	**0.618**	**1.177**
Desktop	GazeFormer	0.228	0.937	0.644	0.924	0.798	7.519	0.821	0.239	0.234	0.650	0.566	0.790
	Chen et al.	0.126	0.846	0.609	0.760	0.687	16.818	0.711	0.185	0.022	0.519	0.502	0.127
	EyeFormer++	0.175	**0.964**	0.608	**0.949**	0.788	11.039	0.828	0.158	0.188	0.616	0.545	0.639
	SeekUI	**0.335**	0.947	**0.640**	0.935	**0.842**	**6.168**	**0.856**	**0.270**	**0.403**	**0.742**	**0.661**	**1.371**
Webpage	GazeFormer	0.236	0.938	0.663	0.934	0.804	7.911	0.819	0.245	0.224	0.663	0.555	0.827
	Chen et al.	0.161	0.857	0.644	0.782	0.721	15.265	0.740	0.203	0.082	0.588	0.558	0.344
	EyeFormer++	0.162	**0.968**	0.621	**0.958**	0.778	12.151	0.823	0.144	0.179	0.617	0.541	0.638
	SeekUI	**0.361**	0.950	**0.690**	0.939	**0.864**	**6.329**	**0.861**	**0.314**	**0.453**	**0.768**	**0.675**	**1.646**

Table 2: Success rates (final fixations within foveal radius of the target) for humans, SeekUI, and baseline models. Human performance sets an upper bound of 70%, reflecting the inherent difficulty of GUI search tasks. SeekUI achieves 62%, closely approaching human performance, while all baselines perform substantially worse (≤18%), indicating their inability to reliably guide search toward targets.

	Ground Truth	**SeekUI**	**GazeFormer**	**Chen et al.**	**EyeFormer++**
Success Rate	70%	**62%**	14%	18%	10%

dynamics but also encodes semantic and spatial guidance sufficient to reliably reach the target.

5.3.2 Saccade Directions. The distribution of saccade directions is shown in Figure 4. Human scanpaths vary systematically with interface type [84]: mobile GUIs are dominated by vertical, downward saccades due to stacked content, while desktop GUIs and webpages show more horizontal transitions, reflecting menus, navigation bars, and side panels.

We quantitatively evaluate models using the Circular Wasserstein Distance between predicted and ground-truth distributions (lower is better). This metric is suited for angular data, accounting for the continuity between 0° and 360° and ensuring boundary-adjacent predictions are properly assessed.

No model fully reproduces these distributions, though limitations differ. EyeFormer++, adapted from free-viewing, overemphasizes vertical saccades, ignoring horizontal exploration, problematic for desktops and webpages. Chen et al.'s model generates excessively long scanpaths, over-predicting saccade counts and yielding the highest Wasserstein distances.

In contrast, GazeFormer and SeekUI align more closely with ground truth, achieving the lowest distances and partially recovering key patterns, such as downward saccades on mobile and horizontal transitions on desktops and webpages. Nonetheless, all models underrepresent the diversity of human strategies.

5.3.3 Saccade Lengths. Figure 5 shows saccade length distributions for humans and models. Human search exhibits a heavy-tailed distribution: most saccades are short for inspecting local elements, punctuated by occasional long saccades to reposition across the interface. This mix reflects a balance of precision and efficiency in navigating complex GUIs.

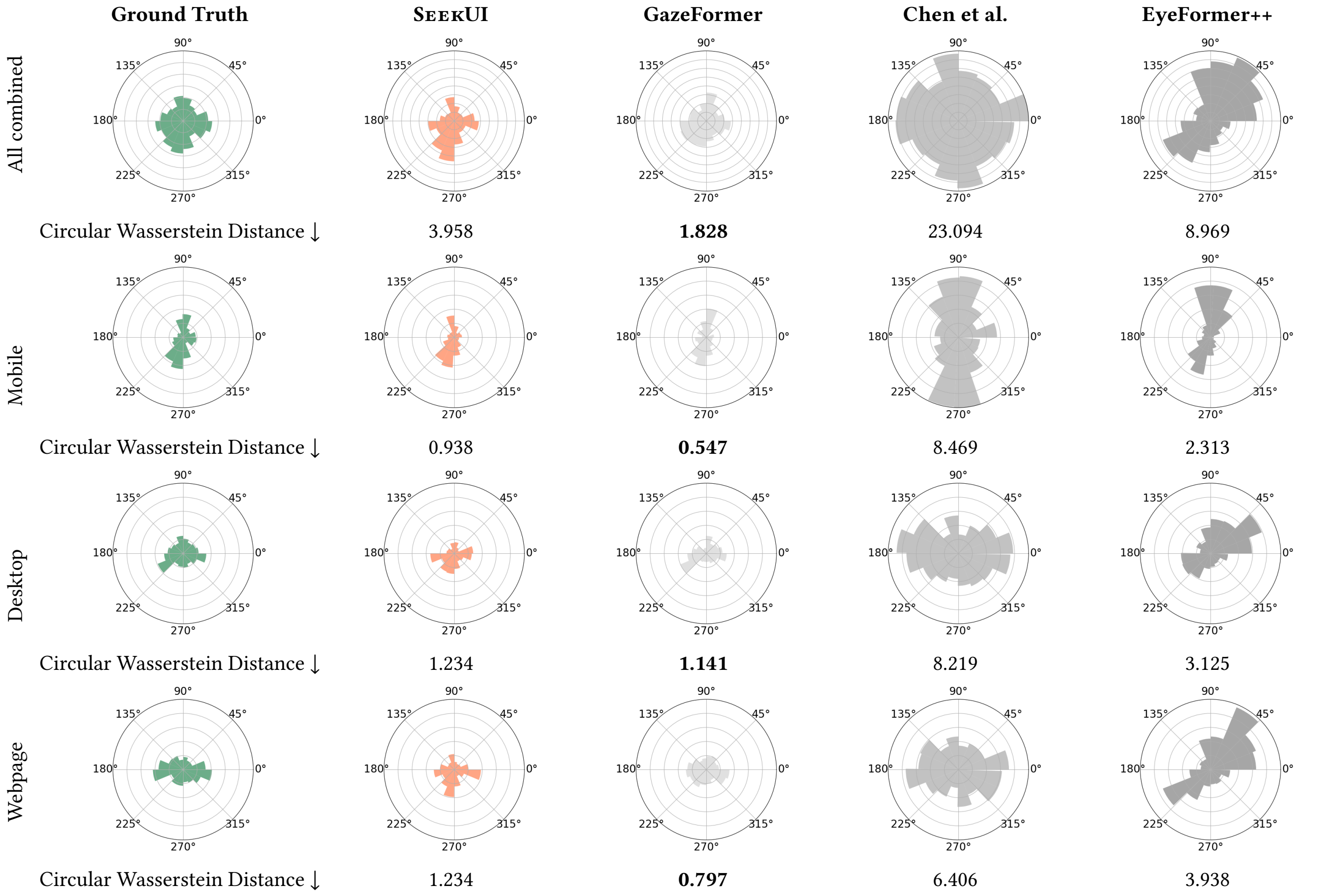


Figure 4: Distribution of saccade directions across interface conditions (all, mobile, desktop, and web). We compare human (ground truth) data with SeekUI and three baselines (GazeFormer, Chen et al., and EyeFormer++). To quantify the similarity between the model predictions and ground truth distributions, we calculate the Circular Wasserstein Distance (lower is better). Human scanpaths show consistent differences across GUI types: downward saccades dominate in mobile GUIs, while horizontal transitions are more common in desktop GUIs and webpages. EyeFormer++ fails to capture these shifts, producing overly vertical predictions across all conditions. Chen et al. generates much longer scanpaths than the ground truth. SeekUI and GazeFormer demonstrate stronger alignment with the ground truth, partially recovering these patterns but still falling short of replicating human saccade direction distributions.

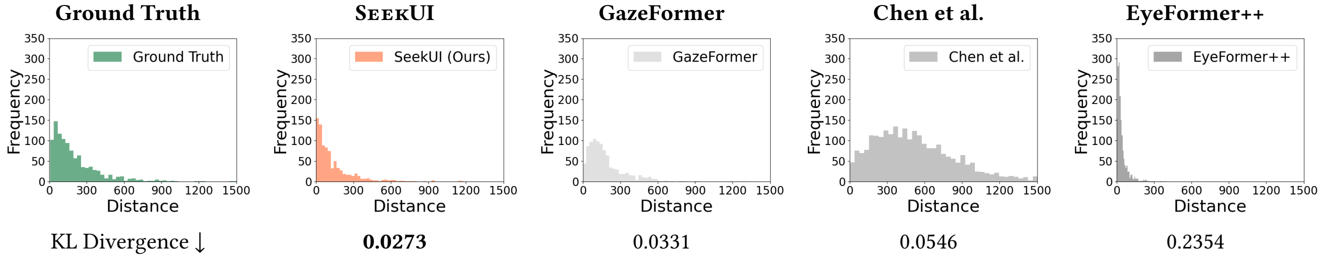


Figure 5: Distribution of saccade lengths for human (ground truth), SeekUI, and baseline models (GazeFormer, Chen et al., EyeFormer++). To evaluate the distribution, we compute the KL Divergence values, which indicate how closely each model matches human behavior. Human scanpaths show a heavy-tailed distribution: short saccades dominate, but occasional long saccades enable rapid shifts across interface regions. SeekUI and GazeFormer best capture this dual-scale pattern, achieving the lowest divergence scores. In contrast, EyeFormer++ misses mid-range transitions and fails to model the heavy tail. Chen et al.'s model produces overly uniform and extended saccades that diverge from human strategies.

We quantify alignment using Kullback-Leibler (KL) Divergence. SeekUI performs best ($KL = 0.0273$), followed by GazeFormer ($KL = 0.0331$), both capturing the dominance of short saccades while producing the occasional long transitions, reflecting human dual-scale strategy. Overall, saccades remain slightly shorter than the human ground truth.

Other baselines struggle: EyeFormer++ has the highest error ($KL = 0.2354$), capturing short movements but missing mid-to-long saccades, producing an overly narrow search range. Chen et al.'s model produces a flatter, overly broad distribution ($KL = 0.0546$), overproducing medium-to-long saccades and failing to replicate the efficiency and local detail of human search.

5.3.4 Visual Complexity vs. Search Time. We analyze how the number of fixations, indicating search time, varies with visual complexity. Figure 6 shows two perspectives: (1) the number of interface elements (left), and (2) visual clutter measured by the Rosenholtz metric [89] (right). Model–human differences are quantified using Mean Squared Error (MSE).

We observed that search time, as reflected by fixation counts, remains relatively stable across GUIs with different numbers of interface elements. This aligns with prior work, which argued that the number of elements is not a reliable predictor of search time [84]. Instead, search time is more strongly influenced by perceptual clutter. As shown in the right plot of Figure 6, the number of fixations increases as visual clutter increases, consistent with findings in visual cognition that clutter is one of the primary drivers of search difficulty [84].

Comparing across models, we find that both SeekUI and GazeFormer successfully replicate the empirical observation that fixation counts remain relatively stable across GUIs with different numbers of elements. This is confirmed by their low MSE scores (2.347 and 2.792, respectively). In contrast, Chen et al. and EyeFormer++ incorrectly predict that the number of fixations decreases as the number of elements increases, which does not reflect real user behavior and results in significantly higher errors (MSE 105.825 and 22.453).

For visual clutter, SeekUI partially captures the increasing trend, achieving the lowest error (MSE 10.437), though the effect is weaker than human data. EyeFormer++ also shows a positive trend but overestimates fixations. GazeFormer and Chen et al. fail to capture clutter effects, with Chen et al. showing the poorest fit (MSE 93.490)..

5.3.5 Guess–Scan–Confirm Pattern. Prior work identifies a consistent behavioral pattern in GUI visual search, the Guess–Scan–Confirm strategy [57, 84]. This three-stage process reflects how users efficiently locate targets: an initial Guess guided by expectations, a Scan across likely candidates, and a Confirm phase to verify the target. We assess whether SeekUI reproduces this pattern. Figure 7 compares fixation heatmaps from ground-truth data and SeekUI predictions, while Table 3 provides quantitative evaluation using Pearson's Correlation (Correlation) [60] and Saliency Map Intersection (Similarity) [90, 96]. Correlation assesses the linear dependence of the spatial distributions to verify structural matching, while Similarity quantifies the direct intersection of probability mass to measure the precise overlap of attention.

Guess. Early fixations in humans and SeekUI concentrate toward the upper-left quadrant, consistent with the well-documented upper-left bias [16, 117], driven by reading habits and interface conventions placing orientation cues (e.g., logos, headers) in this area. SeekUI replicates this expectation-driven bias even when targets lie elsewhere.

Scan: During the Scan phase, human fixations spread across potential target regions, progressively approaching the correct quadrant. SeekUI mirrors this exploration, balancing layout coverage with targeted fixations. Both ground-truth and predicted heatmaps show systematic scanning rather than random movements, although the model occasionally produces slightly sharper focal regions.

Confirm: In the final phase, fixations cluster tightly around the target for verification. SeekUI reproduces this behavior, with late-stage fixation maps showing high similarity to human data. Metrics confirm this: correlations for SeekUI exceed 0.89 in multiple quadrants, whereas baselines like Chen et al. often show low or negative correlations (e.g., -0.18), failing to converge on the target.

Overall, SeekUI captures both spatial fixation patterns and the temporal strategy of human search, aligning with the Guess–Scan–Confirm pattern and demonstrating its ability to model higher-level search organization.

5.4 Ablation Studies

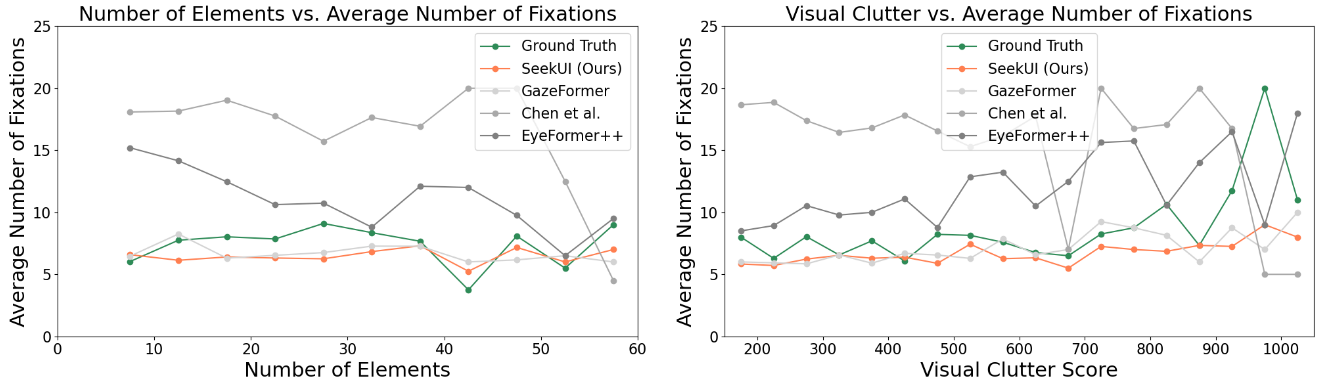
We assess the contribution of each component of SeekUI by removing (1) explanation modeling, (2) RL fine-tuning, (3) both, and additionally comparing to the out-of-the-box VLM. Table 4 summarizes results across all GUIs and for mobile, desktop, and web interfaces.

Three main findings emerge. First, without fine-tuning, the VLM produces illogical and incomplete scanpaths, with low spatial alignment (CC: 0.084) and near-random temporal structure (ScanMatch: 0.095). Its seemingly comparable SED score (6.824) is misleading: the model generates overly short paths, reflected in the very low MultiMatch-Length score (0.505). Visual examples appear in Figure 3.

Second, explanation modeling substantially improves spatial alignment (e.g., CC, NSS, AUC). Comparing the "w/o explanation" variant to the full model shows consistent gains, indicating that including explanatory signals helps the model align gaze predictions with human fixations more effectively.

Third, RL primarily enhances sequence-level coherence. Removing RL decreases ScanMatch (0.347→0.256 overall) and increases SED (5.776→6.952), indicating less structured scanpaths. The full model achieves the best balance across MultiMatch components, confirming that RL strengthens temporal dynamics rather than only static saliency.

These trends hold across interface types. Explanation modeling benefits mobile GUIs most strongly in spatial metrics (CC, NSS), while RL most improves scanpath coherence (ScanMatch). For cluttered webpages, combining both yields the largest overall gains (e.g., CC 0.453, NSS 1.646). Together, these results show that explanation modeling improves spatial fidelity, RL enhances temporal organization, and both are essential for realistic GUI visual search.



	SeekUI	GazeFormer	Chen et al.	EyeFormer++
MSE ↓	**2.347**	2.792	105.825	22.453

	SeekUI	GazeFormer	Chen et al.	EyeFormer++
MSE ↓	**10.437**	11.509	93.490	27.724

Figure 6: Average number of fixations as a function of (left) the number of interface elements and (right) the visual clutter score. We evaluate the model alignment using Mean Squared Error (MSE, lower is better). The number of elements shows little effect on fixation counts, while clutter strongly correlates with increased fixations. SeekUI and GazeFormer reproduce the stable trend with element counts, while Chen et al. and EyeFormer++ predict a decreasing trend. For clutter, SeekUI partially captures the increasing trend and achieves the best quantitative fit, though the effect size is weaker than in human ground-truth data. EyeFormer++ also shows a positive trend with clutter, but overestimates the number of fixations. GazeFormer and Chen et al. fail to capture this effect, with Chen et al. exhibiting the highest MSE error.

Table 3: Quantitative comparison of fixation map alignment across the Guess, Scan, and Confirm stages for targets in each screen quadrant. We report Pearson's Correlation Coefficient (Correlation) and Saliency Map Intersection (Similarity), two widely-used metrics for saliency maps, between model predictions and ground truth. SeekUI achieves the highest scores across most target locations and search phases, indicating it best captures the spatiotemporal dynamics of human search.

Target Quadrant	Methods	Guess		Scan		Confirm	
		Correlation ↑	Similarity ↑	Correlation ↑	Similarity ↑	Correlation ↑	Similarity ↑
Top-Left	GazeFormer	0.79	**0.69**	0.62	**0.74**	0.21	0.48
	Chen et al.	0.49	0.61	0.55	0.66	0.20	0.46
	EyeFormer++	0.73	0.56	0.24	0.47	0.11	0.38
	SeekUI	**0.85**	0.66	**0.84**	0.71	**0.89**	**0.80**
Top-Right	GazeFormer	**0.93**	**0.75**	0.58	0.74	0.51	0.64
	Chen et al.	0.79	0.73	0.38	0.70	0.56	0.55
	EyeFormer++	0.86	0.61	0.01	0.43	0.06	0.46
	SeekUI	0.92	**0.75**	**0.87**	**0.75**	**0.92**	**0.81**
Bottom-Left	GazeFormer	**0.90**	**0.73**	0.41	0.62	0.27	0.45
	Chen et al.	-0.26	0.46	-0.20	0.66	-0.09	0.39
	EyeFormer++	0.85	0.61	0.47	0.42	0.34	0.43
	SeekUI	0.85	0.66	**0.74**	**0.67**	**0.91**	**0.83**
Bottom-Right	GazeFormer	0.76	0.63	**0.38**	**0.69**	0.35	0.59
	Chen et al.	0.47	**0.66**	-0.09	0.80	-0.18	0.50
	EyeFormer++	0.78	0.55	0.24	0.44	0.13	0.43
	SeekUI	**0.81**	0.65	**0.38**	0.63	**0.89**	**0.80**

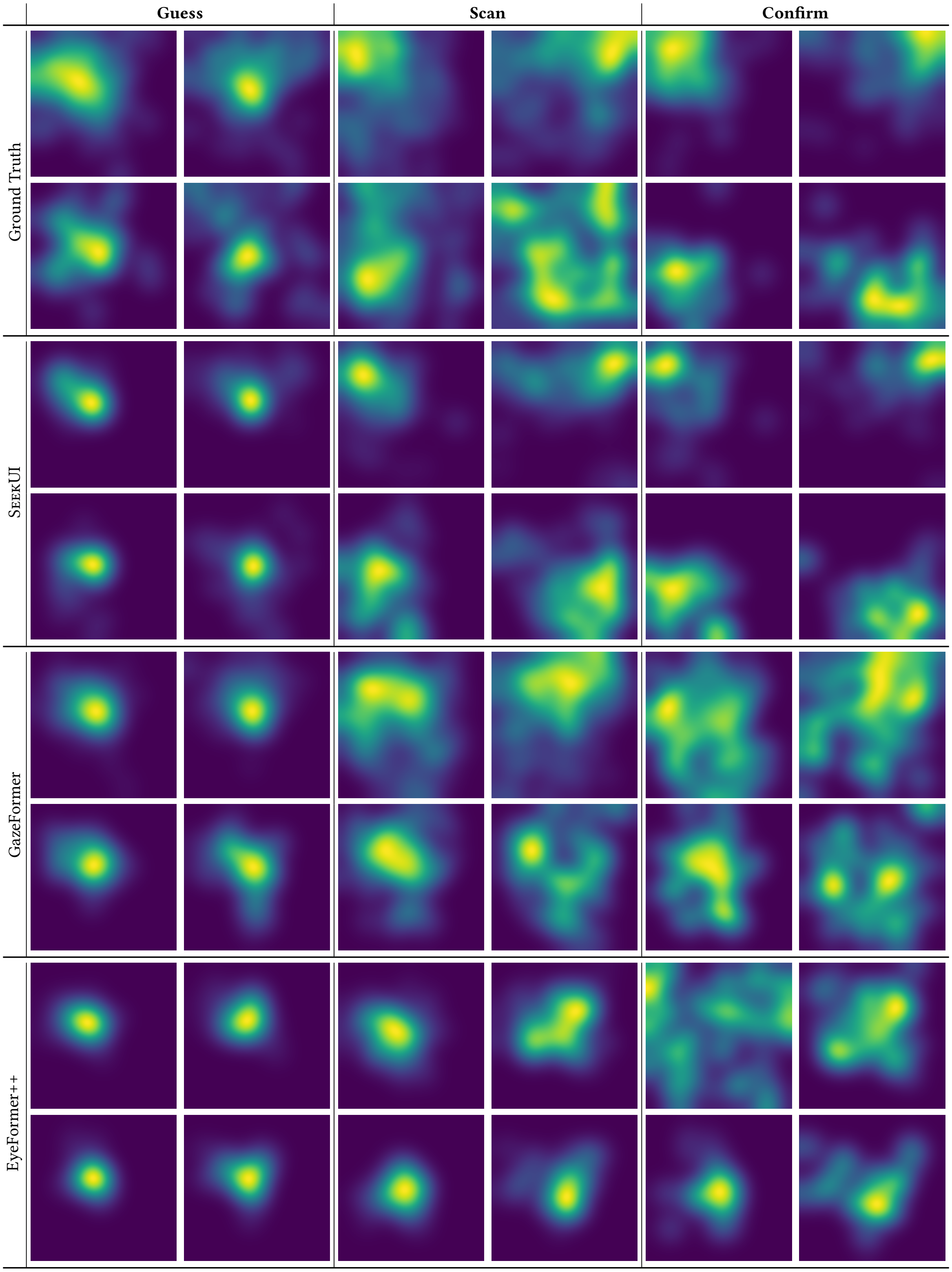


Figure 7: Comparison of fixation heatmaps for ground truth and SeekUI and baselines (GazeFormer and EyeFormer++) across the Guess, Scan, and Confirm stages of visual search. Heatmaps are shown for targets in each screen quadrant, in layouts of top-left, top-right, bottom-left and bottom-right regions. SeekUI mostly successfully reproduces the Guess–Scan–Confirm pattern [57, 84], including the top-left initial bias, systematic scanning of candidate regions, and final confirmation fixations near the target. In contrast, EyeFormer++ fails to capture distinct search phases, exhibiting a static, center-biased distribution throughout the task. In the experiments, we followed the same evaluation procedures as detailed in Putkonen et al. [84] to make a fair comparison.

Table 4: Ablation study evaluating the contribution of explanation modeling and RL to SeekUI's performance. Removing explanation modeling primarily reduces spatial alignment metrics (CC, NSS, AUC), while removing RL mainly decreases scanpath coherence (ScanMatch, SED). The full model consistently achieves the best balance across all metrics and GUI types, demonstrating that both components are necessary for realistic and human-like visual search predictions.

GUI Type	Ablation	ScanMatch ↑	MultiMatch				SED ↓	STDE ↑	SS ↑	CC ↑	AUC ↑	sAUC ↑	NSS ↑
			Vec ↑	Dir ↑	Len ↑	Pos ↑							
All combined	Out-of-the-box VLM	0.095	0.706	0.514	0.505	0.605	6.824	0.692	0.226	0.084	0.541	0.532	0.283
	w/o explanation, RL	0.263	0.933	0.659	0.920	0.809	7.232	0.818	0.283	0.305	0.699	0.616	1.015
	w/o explanation	0.315	**0.948**	**0.677**	0.933	0.837	6.276	0.849	0.295	0.367	0.725	0.629	1.280
	w/o RL	0.256	0.932	0.661	0.919	0.819	6.952	0.826	0.284	0.292	0.688	0.611	0.965
	Full model	**0.347**	0.947	0.677	**0.934**	**0.855**	**5.776**	**0.860**	**0.320**	**0.410**	**0.747**	**0.659**	**1.394**
Mobile	Out-of-the-box VLM	0.099	0.696	0.507	0.504	0.610	5.766	0.698	0.258	0.075	0.545	0.539	0.275
	w/o explanation, RL	0.271	0.929	0.677	0.919	0.827	6.074	0.831	0.316	0.314	0.722	0.611	0.950
	w/o explanation	0.331	0.944	**0.703**	0.923	**0.859**	5.319	0.863	0.341	**0.386**	**0.737**	0.617	**1.253**
	w/o RL	0.275	0.931	0.702	0.916	0.838	5.883	0.839	0.311	0.327	0.702	0.603	1.027
	Full model	**0.346**	**0.944**	0.695	**0.928**	0.857	**4.989**	**0.863**	**0.366**	0.376	0.732	**0.618**	1.177
Desktop	Out-of-the-box VLM	0.118	0.709	0.532	0.509	0.608	7.143	0.695	0.207	0.109	0.545	0.535	0.339
	w/o explanation, RL	0.259	0.937	0.630	0.921	0.796	7.493	0.815	0.252	0.306	0.684	0.607	1.028
	w/o explanation	0.306	**0.949**	**0.653**	**0.937**	0.821	6.493	0.841	0.266	0.363	0.718	0.637	1.290
	w/o RL	0.255	0.930	0.635	0.916	0.816	7.039	0.819	0.268	0.290	0.701	0.624	0.981
	Full model	**0.335**	0.947	0.640	0.935	**0.842**	**6.168**	**0.856**	**0.270**	**0.403**	**0.742**	**0.661**	**1.371**
Webpage	Out-of-the-box VLM	0.067	0.714	0.505	0.501	0.597	7.772	0.683	0.207	0.069	0.531	0.521	0.237
	w/o explanation, RL	0.257	0.933	0.666	0.921	0.802	8.354	0.806	0.275	0.295	0.688	0.600	1.071
	w/o explanation	0.305	**0.953**	0.670	**0.943**	0.828	7.202	0.840	0.267	0.350	0.720	0.612	1.299
	w/o RL	0.233	0.934	0.637	0.925	0.801	8.139	0.817	0.269	0.256	0.661	0.577	0.885
	Full model	**0.361**	0.950	**0.690**	0.939	**0.864**	**6.329**	**0.861**	**0.314**	**0.453**	**0.768**	**0.675**	**1.646**

Table 5: Quantitative alignment analysis between generated explanations and visual scanpaths. The results indicate the correspondence across semantic (WED, IoU), and sequential (BLEU-4, METEOR) dimensions.

Method	WED ↓	IoU ↑	BLEU-4 ↑	METEOR ↑
SeekUI	2.7	69.2	44.5	73.1

5.5 Evaluation of Explanation-Scanpath Alignment

Given the explanation and scanpath produced by SeekUI, we evaluated how well these two outputs align. Direct comparison is difficult because explanations are in natural language while scanpaths are sequences of fixation coordinates. To enable comparison, we instead matched the content elements referenced in the explanations with those traversed in the scanpaths. We randomly sampled 50 explanation–scanpath pairs and manually extracted the corresponding sequences of content elements. Examples of several extracted sequences appear in the Supplementary Materials.

Metrics. To quantify alignment, we compared the extracted sequences using four metrics. We computed the Word Edit Distance (WED), derived from Levenshtein distance, to measure the minimum number of operations needed to transform the explanation sequence into the scanpath sequence, with lower values indicating greater similarity. We also calculated the Intersection over Union (IoU) on the sets of unique content elements to assess semantic overlap. To evaluate sequential consistency, we used BLEU-4 [80] and METEOR [6], which measure n-gram precision and unigram precision–recall, respectively. Both assess whether explanation elements preserve the order of visual exploration.

Results. Table 5 summarizes the results. The IoU of 0.6924 and WED of 2.72 indicate substantial overlap between elements mentioned in explanations and those visited in scanpaths. The BLEU-4 score of 44.51 and METEOR score of 73.12 further demonstrate strong sequential correspondence, showing that explanation structure closely reflects the visual exploration scanpath.

6 Applications

Our model provides practical utility for UI/UX design, especially in early prototyping when eye-tracking data is costly or unavailable. We focus on two applications: evaluating element placement and optimizing full layouts for target-search efficiency.

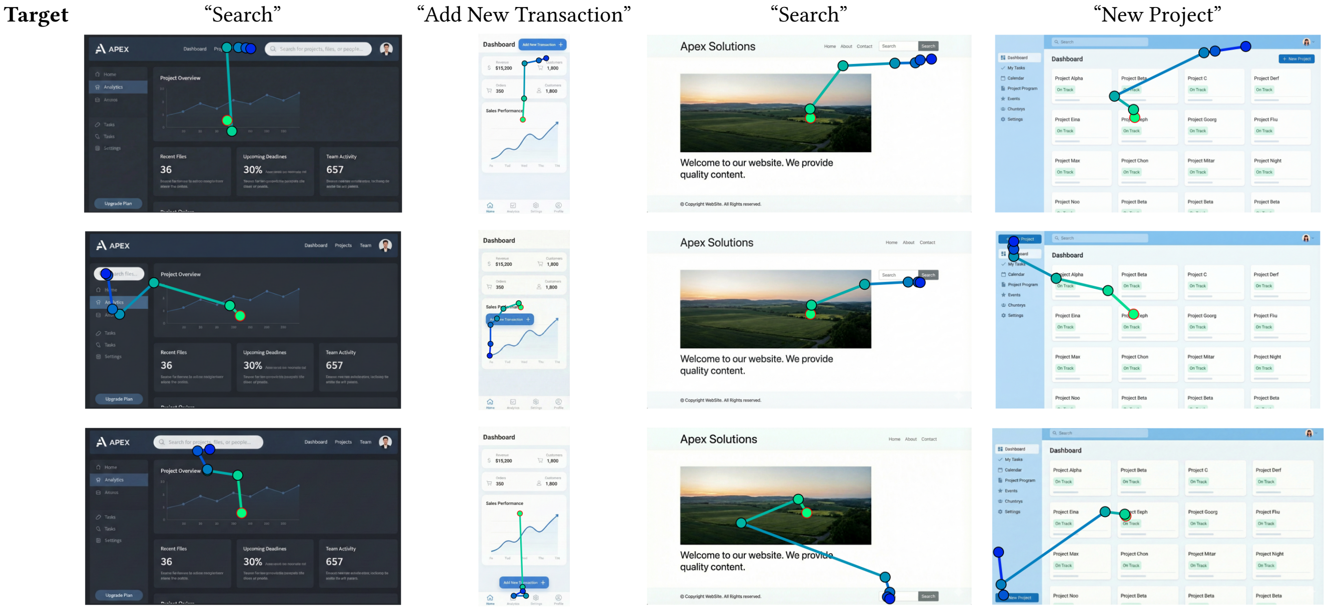


Figure 8: Using SeekUI to evaluate element placement. The three rows show examples of changing the target element's position while keeping all other GUI elements fixed across four screenshots. By repositioning individual GUI elements, designers can observe how predicted scanpaths change and how spatial location affects the likelihood and timing of user attention to each target element.

6.1 Evaluation of Element Placement

SeekUI helps designers assess how the placement of individual GUI elements shapes visual-attention flow. As shown in Figure 8, designers can reposition components (e.g., search bars, menu items, call-to-action buttons, indicators) and immediately generate predicted scanpaths to compare how such changes affect likely viewing behavior.

This enables rapid "what-if" exploration during early design. For example, moving a promotional banner from the sidebar to the header may yield an earlier predicted fixation, while relocating an action button from a dense sidebar to a more isolated area may reduce detours or long saccades. These predictions support iterative refinement so that important elements align with intended visual-search patterns without requiring eye-tracking hardware or repeated user studies.

6.2 Layout Optimization for Target Search Efficiency

Beyond individual elements, SeekUI enables full-layout optimization by predicting how different arrangements influence users' ability to locate key targets. Given a set of elements of interest (e.g., "Search"), SeekUI simulates scanpaths for each target across layout candidates, allowing designers to quickly identify which arrangements minimize search time.

Figure 9 illustrates this workflow. SeekUI (a) predicts scanpaths to each target, (b–c) ranks layouts based on target-specific search difficulty, and (d) supports multi-target optimization by selecting designs that jointly minimize effort. Lower scores correspond to more efficient predicted search behavior. Additional examples are included in the Supplementary Materials.

To quantify performance, SeekUI computes a combined target-search efficiency score, $C_{eff} \in [0, 1]$, where lower values indicate better performance. The score averages three normalized penalties: (1) path efficiency cost C_{path}, (2) fixation number cost C_{fix}, and (3) hit-failure penalty C_{hit}:

$$C_{eff} = (C_{path} + C_{fix} + C_{hit})/3. \tag{13}$$

Path efficiency cost measures deviation from the optimal straight-line route. We compute inefficiency as the ratio of optimal Euclidean distance $d_{optimal}$ to total path length d_{total}:

$$C_{path} = 1 - \frac{d_{optimal}}{d_{total}}. \tag{14}$$

A perfectly direct path yields zero cost.

Fixation number cost reflects increased search time with more fixations. We normalize fixation count N against a threshold ($N_{max} = 20$):

$$C_{fix} = \min\left(\frac{N-1}{N_{max}-1}, 1\right). \tag{15}$$

Hit-failure penalty assigns 0 for successful target detection and 1 otherwise, ensuring failures are fully penalized. Search-success criteria are detailed in Section 5.3.1. Together, these components let SeekUI rank designs by the efficiency and reliability of predicted target search behavior.

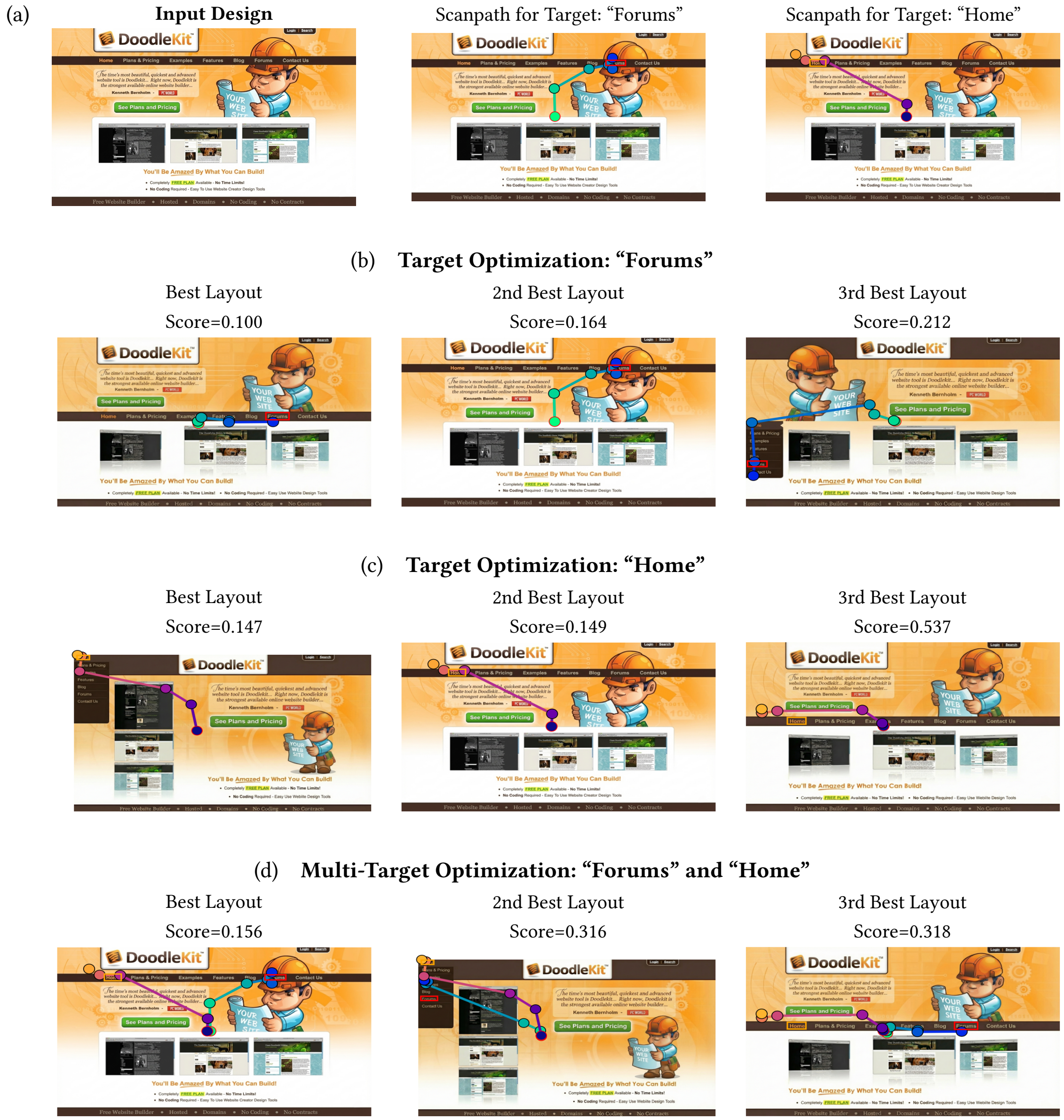


Figure 9: Application of layout optimization for target search efficiency. (a) The original interface and SeekUI's predicted scanpaths for two targets ("Forums" and "Home"). (b) Top-ranked reflowed layouts that minimize predicted search time for "Forums" alone. (c) Top-ranked reflowed layouts that minimize predicted search time for "Home" alone. (d) Layouts selected through joint optimization, minimizing the combined search-efficiency score for both "Forums" and "Home" simultaneously. Lower scores indicate more efficient predicted visual search.

7 Discussion

Our work provides a strong foundation for predicting visual search behaviors on real-world GUIs. We summarize our evaluation findings, discuss the capabilities of VLMs and explanation modeling for scanpath prediction, outline applications, and highlight limitations and future directions.

7.1 Model Accuracy Against Human Behavior

SeekUI not only outperforms baselines in scanpath prediction but reproduces key behavioral signatures of human GUI search. It improves the strongest baseline by 47% in ScanMatch and more than doubles NSS, reflecting superior sequence alignment and fixation distributions across mobile, desktop, and web interfaces. With a 62% success rate, it substantially exceeds GazeFormer (14%), Chen et al. (18%), and EyeFormer++ (10%), approaching the human benchmark of 70%. Behaviorally, it captures consistent saccade direction biases, heavy-tailed saccade lengths, clutter-driven search times, and the Guess–Scan–Confirm strategy. Ablations show explanation modeling improves spatial accuracy (CC, NSS, AUC), while RL enhances temporal coherence (ScanMatch, SED), with the full model achieving the strongest overall balance. These results establish SeekUI as the first human-aligned model of task-driven GUI visual search.

7.2 Capabilities of VLMs and Explanation Modeling

The success of SeekUI demonstrates the potential of VLMs for modeling visual search and understanding GUIs. By integrating textual cues with visual layouts, VLMs can better predict where users look, consistent with evidence that GUI eye movements are influenced by both saliency and semantic content [46, 61].

A central benefit of VLM exploited in SeekUI is its incorporation of explanation modeling. Before producing a scanpath, the model generates a textual rationale describing the search strategy. This design allows SeekUI to predict not only where users are likely to look but also what they are searching for and why they follow particular visual routes. The approach draws inspiration from the two-stream hypothesis of the human visual system [27], in which the ventral stream encodes object identity ("what") and the dorsal stream supports spatial localization and action ("where/how"). By explicitly linking goal semantics (e.g., "the login button") to gaze sequences, explanation modeling provides a bridge between user intent and scanpath dynamics.

Ablation studies show this mechanism improves spatial alignment metrics (CC, NSS, AUC), indicating that explanations anchor predictions to human-like attention patterns. Conceptually, intermediate reasoning steps provide an effective scaffold for goal-directed behavior in multimodal tasks. For practitioners, the textual rationales also enhance interpretability, offering insight into why the model generates particular gaze sequences and giving designers a transparent view of predicted strategies.

7.3 Applications

This research demonstrates that computational models can achieve the accuracy and scalability needed to complement empirical studies of visual search [46, 84, 111]. By acting as a robust "synthetic user", SeekUI enables interface evaluation and optimization without large-scale human studies. First, it can predict whether users efficiently locate key elements, supporting proactive usability testing [28]. Second, it can aid accessibility by simulating how diverse user groups navigate interfaces under varying cognitive or perceptual constraints [23]. Third, it can enhance human–AI collaboration, as systems that search GUIs human-like are more predictable and trustworthy. Finally, integrating such models into generative design workflows can guide layout optimization, building on prior work with saliency and scanpath models [100, 107].

7.4 Limitations and Future Work

While SeekUI advances scanpath prediction for GUI visual search, several limitations remain and open directions for future research.

7.4.1 Addressing Failure Cases. Figure 10 shows representative examples where SeekUI fails to find the correct target. These errors stem primarily from two properties of GUIs: target ambiguity and GUI complexity. In cases of target ambiguity, multiple elements contain the same or highly similar text (e.g., "Graduate" appearing in both navigation and content areas). When faced with such duplicates, SeekUI sometimes fixates on the wrong instance, highlighting its difficulty in disambiguating semantically identical targets. In cases of GUI complexity, the model struggles with cluttered or visually dense interfaces where targets are fine-grained, visually subtle, or surrounded by competing elements. In these settings, the scanpath predictions may drift toward distractors or locate the target. Addressing these failures will require better integration of multimodal cues (text, icons, color, position) and layout reasoning that considers GUI hierarchies. Contextual disambiguation mechanisms, such as task intent or surrounding semantic cues, could help differentiate duplicates. Hierarchical search strategies, parsing the global layout before refining element-level predictions, may further improve robustness in visually complex interfaces.

7.4.2 Personalizing Scanpath Prediction. Our evaluation focuses on population-level performance, which overlooks individual variability in search strategies [44, 46]. Some users scan text first, others prioritize images, reflecting cognitive preferences, cultural conventions, or prior experience. Extending SeekUI to individual-level modeling could simulate personalized scanpaths and support adaptive interface design. Personalized gaze prediction could also enable tailored ad placement, recommendations, or navigation flows; adaptive educational interfaces highlighting challenging content; and more immersive AR/VR experiences aligned with user attention. Modeling scanpath evolution over time would allow GUIs to adapt alongside users, creating interfaces that grow with behavior.

7.4.3 Extending Support for Different Types of Target Cues. While SeekUI generalizes across GUI types, it currently relies mainly on text cues. Real-world interfaces also include icons, logos, color accents, and stylistic cues that guide gaze. Incorporating richer multimodal signals would enable the model to better capture the full semantic and visual context of GUIs. Future work could explore conditioning on text, images, and layout cues simultaneously to improve robustness across diverse interface styles.

7.4.4 Capturing More Complex Search Behaviors. Our experiments were limited to top-down visual search on static screenshots, a simplification of real-world HCI where search is typically dynamic, involving scrolling, dropdowns, or navigating multiple pages [78, 111]. SeekUI currently predicts scanpaths only to immediately visible targets and does not account for interactive actions needed to reveal or access elements. This limits applicability to multi-step tasks. Future work should extend scanpath modeling to dynamic navigation, enabling agents to both look and interact to change the interface state in pursuit of a target.

7.4.5 Data-Driven vs. Cognitive Mechanisms. While SeekUI reproduces complex human behaviors, such as the Guess–Scan–Confirm strategy, it does so by learning statistical regularities rather than simulating the underlying biological or cognitive mechanisms of vision. The model predicts likely human behavior from large datasets

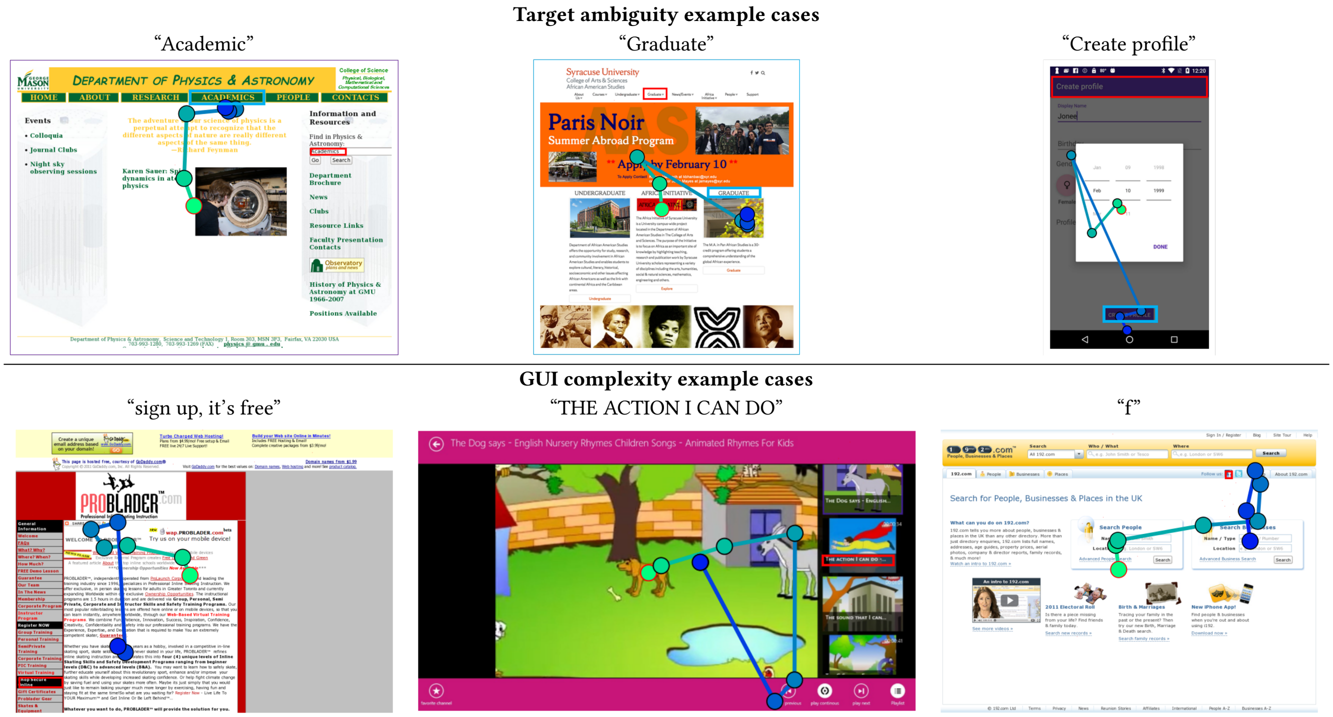


Figure 10: Examples of two most common failure cases of SeekUI: (1) Target Ambiguity, where multiple elements contain the same or highly similar text (the true target is highlighted with a red bounding box, while a distractor with identical text is shown in blue); and (2) GUI Complexity, where cluttered or visually dense layouts lead the model to misidentify the target location.

but does not explicitly model processes like foveated vision, peripheral acuity, or working memory decay. As a result, its scanpaths are plausible and outperform baselines, but it lacks intrinsic explanatory power about human cognitive architecture. Future work could explore hybrid approaches that combine VLM semantic reasoning with biologically constrained cognitive models (e.g., retinal acuity simulations) to bridge statistical prediction and cognitive process modeling.

7.4.6 Human-Centric Validation and Utility. The current training pipeline uses explanations synthesized by a VLM. While we demonstrate strong quantitative alignment between these generated explanations and the resulting visual scanpaths, the study does not include a formal human-in-the-loop validation to confirm that the rationales themselves are qualitatively sound or representative of actual human internal monologues. Future work could involve expert review or crowdsourced assessments to verify the psychological plausibility of these synthesized search strategies. Furthermore, although we formalized a comprehensive suite of metrics to clarify where SeekUI improves upon the state-of-the-art across sequence, spatio-temporal, and distributional dimensions, a remaining limitation is verifying the subjective utility of these improvements. Future research can explore whether the performance gains captured by these metrics translate into tangible benefits for designers during practical interface evaluation and optimization workflows.

8 Conclusion

We propose a novel computational model for predicting scanpaths in visual search on real-world GUIs. Our results show that models developed for free-viewing perform poorly on this task, even when trained with search-specific data, because search behavior strongly depends on how elements are grouped, named, and interact with the target.

We demonstrate that VLMs can effectively capture this interplay by representing both textual and visual aspects of a GUI. When trained with human data using RL, a VLM accurately reproduces not only trajectory-level behavior but also signature HCI effects, such as layout complexity influencing search. This establishes a strong foundation for future work, and we release our code and benchmark for others to build upon.

For practitioners, VLM-based models can serve as "synthetic users" for interface evaluation, accessibility testing, and design optimization. To expand their utility, future work should address individual-level search strategies, incorporate richer multimodal cues, and extend to multi-step interactive tasks.

Acknowledgments

This work was supported by the ERC Advanced Grant (grant number 101141916), the Research Council of Finland (FCAI, grant numbers 328400, 345604, 341763; Subjective Functions, grant number 357578), and the European Innovation Council (SYMBIOTIK project, grant 101071147). We thank the Aalto IT (Triton cluster) and the LUMI supercomputer for providing computational resources. The LUMI supercomputer is owned by the EuroHPC Joint Undertaking and hosted by CSC and the LUMI consortium. We also thank the anonymous reviewers for their valuable feedback throughout the review process.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. Flamingo: a Visual Language Model for Few-Shot Learning. arXiv:2204.14198 [cs.CV]

[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. doi: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] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923* (2025).

[5] Gilles Bailly, Antti Oulasvirta, D.P. Brumby, and Andrew Howes. 2014. Model of Visual Search and Selection Time in Linear Menus. In *CHI '14: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada). Association for Computing Machinery, New York, 3865–3874. doi:10.1145/2556288.2557093

[6] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 65–72.

[7] Amanda Baughan, Tal August, Naomi Yamashita, and Katharina Reinecke. 2020. Keep it Simple: How Visual Complexity and Preferences Impact Search Efficiency on Websites. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–10. doi:10.1145/3313831.3376849

[8] Amanda Baughan, Nigini Oliveira, Tal August, Naomi Yamashita, and Katharina Reinecke. 2021. Do Cross-Cultural Differences in Visual Attention Patterns Affect Search Efficiency on Websites?. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 362, 12 pages. doi:10.1145/3411764.3445519

[9] Stephan A Brandt and Lawrence W Stark. 1997. Spontaneous eye movements during visual imagery reflect the content of the visual scene. *Journal of cognitive neuroscience* 9, 1 (1997), 27–38.

[10] Duncan P. Brumby, Anna L. Cox, Jacqueline Chung, and Byron Fernandes. 2014. How does knowing what you are looking for change visual search behavior?. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 3895–3898. doi:10.1145/2556288.2557064

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

[12] M.D. Byrne. 1993. Using Icons to Find Documents: Simplicity is Critical. In *CHI '93: Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (Amsterdam, The Netherlands). Association for Computing Machinery, New York, 446–453. doi:10.1145/169059.169369

[13] Xianyu Chen, Ming Jiang, and Qi Zhao. 2021. Predicting human scanpaths in visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10876–10885.

[14] Xianyu Chen, Ming Jiang, and Qi Zhao. 2024. Gazexplain: Learning to predict natural language explanations of visual scanpaths. In *European Conference on Computer Vision*. Springer, 314–333.

[15] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. 2023. PaLI: A Jointly-Scaled Multilingual Language-Image Model. arXiv:2209.06794 [cs.CV]

[16] Xin Chen and G.J. Zelinsky. 2006. Real-world visual search is dominated by top-down guidance. *Vision Research* 46, 24 (2006), 4118–4133. doi:10.1016/j.visres.2006.08.008

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

[18] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)* (2023).

[19] Filipe Cristino, Sebastiaan Mathôt, Jan Theeuwes, and Iain D Gilchrist. 2010. ScanMatch: A novel method for comparing fixation sequences. *Behavior research methods* 42, 3 (2010), 692–700.

[20] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. arXiv:2305.06500 [cs.CV]

[21] Richard Dewhurst, Marcus Nyström, Halszka Jarodzka, Tom Foulsham, Roger Johansson, and Kenneth Holmqvist. 2012. It depends on how you look at it: Scanpath comparison in multiple dimensions with MultiMatch, a vector-based approach. *Behavior research methods* 44 (2012), 1079–1100.

[22] Arturo Deza, Jeffrey R. Peters, Grant S. Taylor, Amit Surana, and Miguel P. Eckstein. 2017. Attention Allocation Aid for Visual Search. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 220–231. doi:10.1145/3025453.3025834

[23] Parvin Emami, Yue Jiang, Zixin Guo, and Luis A Leiva. 2024. Impact of Design Decisions in Scanpath Modeling. *Proceedings of the ACM on Human-Computer Interaction* 8, ETRA (2024), 1–16.

[24] Sarah P. Everett and Michael D. Byrne. 2004. Unintended effects: varying icon spacing changes users' visual search strategy. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vienna, Austria) *(CHI '04)*. Association for Computing Machinery, New York, NY, USA, 695–702. doi:10.1145/985692.985780

[25] Tom Foulsham and Geoffrey Underwood. 2008. What can saliency models predict about eye movements? Spatial and sequential aspects of fixations during encoding and recognition. *Journal of vision* 8, 2 (2008), 6–6.

[26] Aryan Garg, Yue Jiang, and Antti Oulasvirta. 2025. Controllable gui exploration. *arXiv preprint arXiv:2502.03330* (2025).

[27] Melvyn A Goodale and A David Milner. 1992. Separate visual pathways for perception and action. *Trends in neurosciences* 15, 1 (1992), 20–25.

[28] Michael Grahame, Jason Laberge, and C.T. Scialfa. 2004. Age Differences in Search of Web Pages: The Effects of Link Size, Link Number, and Clutter. *Human Factors* 46, 3 (2004), 385–398. doi:10.1518/hfes.46.3.385.50404 PMID 15573540.

[29] Zixin Guo, Jiayang Sun, Tzu-Jui Julius Wang, Abduljalil Radman, Selen Pehlivan, Min Cao, and Jorma Laaksonen. 2025. Learning to Describe Implicit Changes: Noise-robust Pre-training for Image Difference Captioning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*. Association for Computational Linguistics, 10125–10145.

[30] Zixin Guo, Tzu-Jui Wang, and Jorma Laaksonen. 2022. CLIP4IDC: CLIP for Image Difference Captioning. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*. 33–42.

[31] Zixin Guo, Tzu-Jui Julius Wang, Selen Pehlivan, Abduljalil Radman, and Jorma Laaksonen. 2023. PiTL: Cross-modal Retrieval with Weakly-supervised Vision-language Pre-training via Prompting. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2261–2265.

[32] T. Halverson and A.J. Hornof. 2004. Local Density Guides Visual Search: Sparse Groups are First and Faster. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 48, 16 (2004), 1860–1864. doi:10.1177/154193120404801615

[33] Tim Halverson and A.J. Hornof. 2011. A Computational Model of "Active Vision" for Visual Search in Human–Computer Interaction. *Human–Computer Interaction* 26, 4 (2011), 285–314. doi:10.1080/07370024.2011.625237

[34] Tim Halverson and Anthony J. Hornof. 2007. A minimal model for predicting visual search in human-computer interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 431–434. doi:10.1145/1240624.1240693

[35] Lena Hegemann, Yue Jiang, Joon Gi Shin, Yi-Chi Liao, Markku Laine, and Antti Oulasvirta. 2023. Computational Assistance for User Interface Design: Smarter Generation and Evaluation of Design Ideas. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–5.

[36] John Henderson. 2005. Introduction to real-world scene perception. *Visual Cognition* 12, 6 (2005), 849–851.

[37] J.M. Henderson and Fernanda Ferreira. 2004. *Scene perception for psycholinguists.* Psychology Press, 1–58.

[38] Corey Holland, Oleg Komogortsev, and Dan Tamir. 2012. Identifying usability issues via algorithmic detection of excessive visual search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 2943–2952. doi:10.1145/2207676.2208703

[39] A.J. Hornof. 2004. Cognitive Strategies for the Visual Search of Hierarchical Computer Displays. *Human–Computer Interaction* 19, 3 (2004), 183–223. doi:10.1207/s15327051hci1903_1

[40] 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.

[41] Yue Jiang. 2024. Computational Representations for Graphical User Interfaces. In *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems (CHI EA '24)*.

[42] Yue Jiang. 2025. Computational representations for user interfaces. (2025).

[43] Yue Jiang, Ruofei Du, Christof Lutteroth, and Wolfgang Stuerzlinger. 2019. ORC Layout: Adaptive GUI Layout with OR-Constraints. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article 413, 12 pages. doi:10.1145/3290605.3300643

[44] Yue Jiang, Zixin Guo, Hamed Rezazadegan Tavakoli, Luis A. Leiva, and Antti Oulasvirta. 2024. EyeFormer: Predicting Personalized Scanpaths with Transformer-Guided Reinforcement Learning. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 47, 15 pages. doi:10.1145/3654777.3676436

[45] Yue Jiang, Luis A Leiva, Hamed Rezazadegan Tavakoli, Paul RB Houssel, Julia Kylmälä, and Antti Oulasvirta. 2023. UEyes: An Eye-Tracking Dataset across User Interface Types. In *Workshop Paper at the 2023 CHI Conference on Human Factors in Computing Systems*.

[46] Yue Jiang, Luis A Leiva, Hamed Rezazadegan Tavakoli, Paul RB Houssel, Julia Kylmälä, and Antti Oulasvirta. 2023. Ueyes: Understanding visual saliency across user interface types. In *Proceedings of the 2023 CHI conference on human factors in computing systems*. 1–21.

[47] Yue Jiang, Yuwen Lu, Clara Kliman-Silver, Christof Lutteroth, Toby Jia-Jun Li, Jeffrey Nichols, and Wolfgang Stuerzlinger. 2024. Computational Methodologies for Understanding, Automating, and Evaluating User Interfaces. In *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems (CHI EA '24)*.

[48] Yue Jiang, Yuwen Lu, Christof Lutteroth, Toby Jia-Jun Li, Jeffrey Nichols, and Wolfgang Stuerzlinger. 2023. The future of computational approaches for understanding and adapting user interfaces. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–5.

[49] Yue Jiang, Yuwen Lu, Jeffrey Nichols, Wolfgang Stuerzlinger, Chun Yu, Christof Lutteroth, Yang Li, Ranjitha Kumar, and Toby Jia-Jun Li. 2022. Computational approaches for understanding, generating, and adapting user interfaces. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 1–6.

[50] Yue Jiang, Christof Lutteroth, Rajiv Jain, Christopher Tensmeyer, Varun Manjunatha, Wolfgang Stuerzlinger, and Vlad I Morariu. 2024. FlexDoc: Flexible Document Adaptation through Optimizing both Content and Layout. In *2024 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 217–222.

[51] Yue Jiang, Eldon Schoop, Amanda Swearngin, and Jeffrey Nichols. 2025. Iluvui: Instruction-tuned language-vision modeling of uis from machine conversations. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*. 861–877.

[52] Yue Jiang, Wolfgang Stuerzlinger, and Christof Lutteroth. 2021. ReverseORC: Reverse engineering of resizable user interface layouts with or-constraints. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–18.

[53] Yue Jiang, Wolfgang Stuerzlinger, Matthias Zwicker, and Christof Lutteroth. 2020. Orcsolver: An efficient solver for adaptive gui layout with or-constraints. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.

[54] Yue Jiang, Changkong Zhou, Vikas Garg, and Antti Oulasvirta. 2024. Graph4gui: Graph neural networks for representing graphical user interfaces. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–18.

[55] J.P.P. Jokinen, Sayan Sarcar, Antti Oulasvirta, Chaklam Silpasuwanchai, Zhenxin Wang, and Xiangshi Ren. 2017. Modelling learning of new keyboard layouts. In *CHI '17: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 4203–4215.

[56] 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.

[57] M.A. Just and P.A. Carpenter. 1976. Eye fixations and cognitive processes. *Cognitive Psychology* 8, 4 (Oct. 1976), 441–480.

[58] David E. Kieras and Anthony J. Hornof. 2014. Towards accurate and practical predictive models of active-vision-based visual search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 3875–3884. doi:10.1145/2556288.2557324

[59] 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).

[60] 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.

[61] 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 *Proceedings of the International conference on human-computer interaction with mobile devices and services*. 1–12.

[62] Gang Li and Yang Li. 2022. Spotlight: Mobile UI Understanding using Vision-Language Models with a Focus. *ArXiv* abs/2209.14927 (2022). https://api.semanticscholar.org/CorpusID:252595735

[63] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*. PMLR, 19730–19742.

[64] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*. PMLR, 12888–12900.

[65] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems* 34 (2021), 9694–9705.

[66] Jonathan Ling and Paul van Schaik. 2007. The influence of line spacing and text alignment on visual search of web pages. *Displays* 28, 2 (2007), 60–67. doi:10.1016/j.displa.2007.04.003

[67] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. https://llava-vl.github.io/blog/2024-01-30-llava-next/

[68] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual Instruction Tuning.

[69] Weilin Liu, Yaqin Cao, and R.W. Proctor. 2021. How do app icon color and border shape influence visual search efficiency and user experience? Evidence from an eye-tracking study. *International Journal of Industrial Ergonomics* 84, Article 103160 (2021). doi:10.1016/j.ergon.2021.103160

[70] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[71] Yuwen Lu, Yue Jiang, Tiffany Knearem, Clara E Kliman-Silver, Christof Lutteroth, Jeffrey Nichols, and Wolfgang Stuerzlinger. 2025. Designing and Developing User Interfaces with AI: Advancing Tools, Workflows, and Practices. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–7.

[72] Daniel Martin, Diego Gutierrez, and Belen Masia. 2022. A probabilistic time-evolving approach to scanpath prediction. *arXiv preprint arXiv:2204.09404* (2022).

[73] Daniel Martin, Ana Serrano, Alexander W Bergman, Gordon Wetzstein, and Belen Masia. 2022. Scangan360: A generative model of realistic scanpaths for 360 images. *IEEE Transactions on Visualization and Computer Graphics* 28, 5 (2022), 2003–2013.

[74] Aliaksei Miniukovich and Antonella De Angeli. 2015. Computation of Interface Aesthetics. In *CHI '15: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea). Association for Computing Machinery, New York, 1163–1172.

[75] Silviu Minut and Sridhar Mahadevan. 2001. A reinforcement learning model of selective visual attention. In *Proceedings of the fifth international conference on Autonomous agents*. 457–464.

[76] Sounak Mondal, Zhibo Yang, Seoyoung Ahn, Dimitris Samaras, Gregory Zelinsky, and Minh Hoai. 2023. Gazeformer: Scalable, effective and fast prediction of goal-directed human attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1441–1450.

[77] Saul B Needleman and Christian D Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology* 48, 3 (1970), 443–453.

[78] M.B. Neider and G.J. Zelinsky. 2008. Exploring set size effects in scenes: Identifying the objects of search. *Visual Cognition* 16, 1 (2008), 1–10. doi:10.1080/13506280701381691

[79] Dimitri Ognibene, Christian Balkenius, and Gianluca Baldassarre. 2008. A reinforcement-learning model of top-down attention based on a potential-action map. In *The Challenge of Anticipation: A Unifying Framework for the Analysis and Design of Artificial Cognitive Systems*. Springer, 161–184.

[80] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the*

40th annual meeting of the Association for Computational Linguistics. 311–318.

[81] Yi-Hao Peng, Faria Huq, Yue Jiang, Jason Wu, Xin Yue Li, Jeffrey P Bigham, and Amy Pavel. 2024. Dreamstruct: Understanding slides and user interfaces via synthetic data generation. In *European Conference on Computer Vision*. Springer, 466–485.

[82] 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.

[83] Peter Pirolli, Stuart K. Card, and Mija M. Van Der Wege. 2001. Visual information foraging in a focus + context visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, USA) *(CHI '01)*. Association for Computing Machinery, New York, NY, USA, 506–513. doi:10.1145/365024.365337

[84] Aini Putkonen, Yue Jiang, Jingchun Zeng, Olli Tammilehto, Jussi PP Jokinen, and Antti Oulasvirta. 2025. Understanding visual search in graphical user interfaces. *International Journal of Human-Computer Studies* 199 (2025), 103483.

[85] Mengyu Qiu, Quan Rong, Dong Liang, and Huawei Tu. 2023. Visual ScanPath Transformer: Guiding Computers to See the World. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 223–232.

[86] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020 [cs.CV] https://arxiv.org/abs/2103.00020

[87] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732* (2015).

[88] 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. doi:10.1016/j.imavis.2013.06.006

[89] Ruth Rosenholtz, Yuanzhen Li, and Lisa Nakano. 2007. Measuring visual clutter. *Journal of Vision* 7, 2, Article 17 (08 2007). doi:10.1167/7.2.17

[90] 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.

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

[92] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300* (2024).

[93] Sruti Srinivasa Ragavan, Sandeep Kaur Kuttal, Charles Hill, Anita Sarma, David Piorkowski, and Margaret Burnett. 2016. Foraging Among an Overabundance of Similar Variants. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 3509–3521. doi:10.1145/2858036.2858469

[94] Xiangjie Sui, Yuming Fang, Hanwei Zhu, Shiqi Wang, and Zhou Wang. 2023. ScanDMM: A Deep Markov Model of Scanpath Prediction for 360deg Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6989–6999.

[95] Wanjie Sun, Zhenzhong Chen, and Feng Wu. 2019. Visual scanpath prediction using IOR-ROI recurrent mixture density network. *IEEE transactions on pattern analysis and machine intelligence* 43, 6 (2019), 2101–2118.

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

[97] Maryam Taeb, Amanda Swearngin, Eldon Schoop, Ruijia Cheng, Yue Jiang, and Jeffrey Nichols. 2024. Axnav: Replaying accessibility tests from natural language. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–16.

[98] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599* (2025).

[99] Leong-Hwee Teo, Bonnie John, and Marilyn Blackmon. 2012. CogTool-Explorer: a model of goal-directed user exploration that considers information layout. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 2479–2488. doi:10.1145/2207676.2208414

[100] Nađa Terzimehić, Renate Häuslschmid, Heinrich Hussmann, and m.c. schraefel. 2019. A Review & Analysis of Mindfulness Research in HCI: Framing Current Lines of Research and Future Opportunities. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3290605.3300687

[101] Kashyap Todi, Jussi Jokinen, Kris Luyten, and Antti Oulasvirta. 2019. Individualising Graphical Layouts with Predictive Visual Search Models. *ACM Transactions on Interactive Intelligent Systems* 10, 1, Article 9 (Aug. 2019), 24 pages. doi:10.1145/3241381

[102] A.K. Trapp and Carolin Wienrich. 2018. App icon similarity and its impact on visual search efficiency on mobile touch devices. *Cognitive Research: Principles and Implications* 3, 1, Article 39 (2018).

[103] A.M. Treisman and Garry Gelade. 1980. A feature-integration theory of attention. *Cognitive Psychology* 12, 1 (1980), 97–136.

[104] Yuan-Chi Tseng and Andrew Howes. 2008. The adaptation of visual search strategy to expected information gain. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 1075–1084. doi:10.1145/1357054.1357221

[105] Wei Wang, Cheng Chen, Yizhou Wang, Tingting Jiang, Fang Fang, and Yuan Yao. 2011. Simulating human saccadic scanpaths on natural images. In *CVPR 2011*. 441–448. doi:10.1109/CVPR.2011.5995423

[106] Yao Wang, Andreas Bulling, et al. 2023. Scanpath prediction on information visualisations. *IEEE Transactions on Visualization and Computer Graphics* (2023).

[107] Yao Wang, Yue Jiang, Zhiming Hu, Constantin Ruhdorfer, Mihai Bâce, and Andreas Bulling. 2024. VisRecall++: Analysing and predicting visualisation recallability from gaze behaviour. *Proceedings of the ACM on Human-Computer Interaction* 8, ETRA (2024), 1–18.

[108] 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*. 3184–3193. doi:10.1109/CVPR.2018.00336

[109] J.M. Wolfe. 2021. Guided Search 6.0: An updated model of visual search. *Psychonomic Bulletin & Review* 28, 4 (2021), 1060–1092.

[110] J.M. Wolfe and T.S. Horowitz. 2004. What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience* 5, 6 (2004), 495–501.

[111] J.M. Wolfe, E.M. Palmer, and T.S. Horowitz. 2010. Reaction time distributions constrain models of visual search. *Vision Research* 50, 14 (2010), 1304–1311. doi:10.1016/j.visres.2009.11.002

[112] 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. doi:10.1109/TIP.2019.2897966

[113] Mai Xu, Yuhang Song, Jianyi Wang, MingLang Qiao, Liangyu Huo, and Zulin Wang. 2018. Predicting head movement in panoramic video: A deep reinforcement learning approach. *IEEE transactions on pattern analysis and machine intelligence* 41, 11 (2018), 2693–2708.

[114] Zhibo Yang, Lihan Huang, Yupei Chen, Zijun Wei, Seoyoung Ahn, Gregory Zelinsky, Dimitris Samaras, and Minh Hoai. 2020. Predicting Goal-Directed Human Attention Using Inverse Reinforcement Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[115] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4651–4659.

[116] Arianna Yuan and Yang Li. 2020. Modeling Human Visual Search Performance on Realistic Webpages Using Analytical and Deep Learning Methods. In *CHI '20: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, 12 pages. doi:10.1145/3313831.3376870

[117] G.J. Zelinsky. 1996. Using Eye Saccades to Assess the Selectivity of Search Movements. *Vision Research* 36, 14 (1996), 2177–2187. doi:10.1016/0042-6989(95)00300-2