# GLAZE: A Gaze-Language Benchmark for Grounding Human Gaze on Web User Interfaces

**Yiran Ma, Yi-Hao Peng, Jiayang Shi, Isha Swaminathan, Aritra Dutta[†], Andrew Begel**
Carnegie Mellon University
{erynm,yihaop,jiayangs,iswamina,aritraad,abegel}@andrew.cmu.edu

## Abstract

Eye gaze offers multimodal language models a grounding signal for understanding user interfaces, such as inferring which element a user intends to interact with. But natural gaze on dense interfaces rarely lands on a single element box or semantic group. Users scan between distant regions and revisit earlier elements, and their attention often spreads across different parts of the layout. Existing research either studies UI eye tracking without aligned language supervision or uses gaze to guide models in natural image and egocentric settings. We introduce GLAZE[1], the first benchmark for gaze-language grounding on web user interfaces. GLAZE is built from a think-aloud study in which 27 participants described real-world webpages while an eye tracker recorded their gaze. The dataset pairs more than 10,000 UI elements across 176 webpages with time-aligned gaze and verbal descriptions. The benchmark provides a model with a screenshot and a gaze visualization and evaluates whether it can identify the interface content or function the user is focusing on. We evaluate seven open and proprietary multimodal language models across five gaze visualizations. Current models often identify the general region being viewed but miss the precise attended elements and their function. Heatmaps and scanpaths are most effective for different UI element types, yet overall performance remains limited across visualization methods. Our benchmark can be useful for future research on UI agents, gaze-based accessibility tools, and automated interface usability analysis.

## 1 Introduction

Eye gaze carries information about a user's attention, intent, and cognitive state (Pani & Yang, 2025). It is also becoming easy to capture, from ordinary webcams (Papoutsaki et al., 2017) to AR/VR headsets (Plopski et al., 2023) and smart glasses (Gao et al., 2024), so gaze-aware systems no longer depend on specialized lab equipment. Recent work feeds gaze into multimodal language models (MLLMs) during training or at inference and reports gains in egocentric activity understanding (Pani & Yang, 2025; Mazzamuto et al., 2025), visual question answering on smart glasses (Konrad et al., 2024), and assistive interaction for people with speech and motor impairments (ElMallah et al., 2026).

Most existing multimodal modeling work has studied gaze on natural and egocentric images rather than user interfaces (UIs). Yet UI elements such as buttons, menus, and text fields are discrete and functionally distinct, so a fixation can point to a specific action a user might take. Elements also sit inside groups, layout hierarchies, and flows that span screens (Wu et al., 2023; Peng et al., 2023), and gaze patterns across them can reveal intentions that no single fixation shows. Multimodal UI agents (Hong et al., 2024; Wu et al., 2025; Peng et al., 2025b; Wu et al., 2026), voice-controlled assistants, and automated usability and accessibility analysis (Abascal et al., 2019; Mowar et al., 2025; Paiva et al., 2021) could all use such information. At the same time, UIs make gaze hard to interpret. Screens densely pack text,

[†]Work done while at Carnegie Mellon University.
[1]Project website: https://eryn-y.github.io/glaze/

images, and icons at different scales, so a small shift in gaze can land on a functionally different element (Feit et al., 2017). Gaze is also task-driven, and natural gaze often drifts, revisits earlier regions, or spans several fixation clusters. Model attention can diverge from human attention (Carrasco et al., 2026), while model performance can depend strongly on how gaze is presented (Yan et al., 2024). We therefore ask whether, given a screenshot and a user's gaze on it, current models can infer what the user is attending to.

Existing eye-tracking datasets for UIs, such as UEyes (Jiang et al., 2023), record where gaze fell but not what the user was attending to, so they do not provide semantic labels for evaluating a model's interpretation. We introduce GLAZE, the first gaze-language grounding benchmark for web UIs. GLAZE is built from a think-aloud study in which 27 participants with UI/UX design experience described real-world webpage screenshots while an eye tracker recorded their gaze. We align the narration with the recorded gaze to label the target of each gaze segment, typically a single element or a small set of related ones. The resulting dataset pairs more than 10,000 UI elements across 176 webpage screenshots with gaze segments and verbal descriptions. The benchmark task gives a model a screenshot and one gaze segment and asks for the interface content or function the participant attended to at that moment.

We evaluate seven open and proprietary MLLMs across five gaze visualizations: raw gaze, scanpath, heatmap, convex hull, and element bounding boxes. Models perform best with heatmaps, yet even then they are fully correct only 19.8% of the time and consistently predict regions larger than the user's actual focus. Models also perform much better on spatially extended elements (e.g., tables, grids) than on compact elements (e.g., icons, buttons). On those extended layouts, models perform better with scanpaths than with heatmaps. In summary, we contribute a publicly released benchmark of aligned gaze, verbal descriptions, and UI screenshots; a systematic evaluation of seven MLLMs across five gaze visualizations; and an analysis of gaze behavior across 21 UI element categories.

## 2 Related Work

### 2.1 Gaze Benchmarks and Datasets

Many gaze benchmarks study natural images or video rather than interfaces. GazeXplain attaches natural language explanations to scanpaths (Chen et al., 2024), and GLAM decodes the category of a visual search target from them (Mondal et al., 2025). Both operate on natural images, and neither records the viewer's own account of what they attended to. EgoGazeVQA and StreamGaze build gaze-conditioned QA tasks for egocentric and streaming video (Peng et al., 2026a; Lee et al., 2025), but their question-answer pairs are generated by models rather than reported by the person whose gaze was recorded.

UEyes is one of the few datasets that record gaze on interfaces (Jiang et al., 2023). It covers four interface types, but its short free-viewing traces include no verbal descriptions. Outside the UI domain, *Localized Narratives* and *Video Localized Narratives* synchronize language with a pointing signal from the same annotator (Pont-Tuset et al., 2020; Voigtlaender et al., 2023), and REFLACX pairs expert gaze with dictated reports in radiology (Lanfredi et al., 2022). Our benchmark brings this style of synchronized annotation to user interfaces and uses it to evaluate whether a model can recover what a user reports attending to, not only where gaze falls. That distinction matters on graphical interfaces, where elements a few pixels apart can have entirely different functions.

### 2.2 Gaze as Input to Multimodal Models

Gaze has also been used as a direct input to multimodal models. At inference time, Voila-A aligns MLLM attention with user gaze (Yan et al., 2024), MindEye-OmniAssist infers open-ended assistive intent from gaze (Zhang et al., 2025), and MICA grounds task assistance in gaze and speech from a single demonstration (Sarch et al., 2025). On the training side, EyeTrans merges human and model attention for code summarization (Zhang et al., 2024), gaze-regularized MLLMs improve egocentric understanding (Pani & Yang, 2025), Thinking

with Gaze adds gaze tokens for medical reasoning (Li et al., 2026), and reward models have been trained with gaze for RLHF-style alignment (Galliamov et al., 2025).

These systems all assume the model understands the gaze it receives. Our benchmark tests that assumption on user interfaces, where the model first has to work out which on-screen element the person is looking at. Prior studies encode gaze as scanpaths, heatmaps, and other visual overlays (Chen et al., 2024; Yan et al., 2024; Peng et al., 2026a), and model performance varies between formats (Yan et al., 2024). Beyond gaze, drawing marks such as boxes or circles directly on the image can point a multimodal model at a region (Lee et al., 2024; Lin et al., 2025). We present gaze visualizations to models in the same way and compare five common methods on dense interfaces.

## 3 Benchmark

Our benchmark evaluates whether a multimodal language model (MLLM) can infer which UI element a user is actively attending to, given a screenshot of the interface and a visualization of the user's gaze on that screenshot. Ground truth comes from the participant's own concurrent verbal description of what they are looking at. A coverage-based evaluation scheme then compares the model's prediction against this ground truth, measuring whether the predicted region correctly identifies the target (Section 4.4).

### 3.1 Dataset Construction

We recruited 27 participants with UI/UX experience. Each participant completed a 90-minute session, viewing 4–6 webpage screenshots depending on their pace of description. Following the established think-aloud protocol (Van Someren et al., 1994), participants described the screenshot while concurrently verbalizing the interface elements they attended to. Participants were instructed to describe as many visual elements as they considered worth mentioning. No description order was prescribed; participants often proceeded roughly from top to bottom and left to right, following the visual hierarchy, page flow, or functional groupings. During the experiment, their speech, gaze, mouse position, and screen content were recorded. Familiarity with interface conventions allows participants to narrate fluidly and with minimal interruption, yielding higher-quality gaze–speech alignment. Their demographic and UI-background are reported in Appendix A.1. The aligned gaze and speech data totals 18 hours and 22 minutes, spanning 131 unique UI screenshots and 7,137 UI elements, with an average description time of 8.5 minutes per screenshot. To supplement the data, the experimenter, who also had UI/UX experience, qualitatively analyzed what participants typically covered on UI screenshots and annotated 45 additional unique screenshots and 3,280 additional elements, in total of 9 hours and 13 minutes. The final benchmark comprises 176 unique screenshots, 10417 UI elements, and 27.5 hours of gaze-speech data. Aggregate behavioral comparisons between the participant- and experimenter-collected subsets are reported in Appendix A.3.

The screenshots are all $1920 \times 1200$ px. The experimenter manually selected the screenshots from real, in-the-wild websites to ensure they cover a wide range of topics (from music to gameplay to travel) and layouts. Detailed breakdowns of website and UI categories are in Figure A.2 in Appendix A.3. Gaze data was collected using a Tobii Pro Fusion remote eye tracker sampling at 60 Hz. The eye tracker was attached at the bottom of a 24.1-inch Dell monitor. Every participant's session was first calibrated using Tobii's calibration software and instructed to keep their head stable throughout the session. If a participant moved too much, we reran their calibration. Raw gaze points were then classified into fixations and saccades using the I-VT (Velocity-Threshold Identification) filter with Tobii's recommended defaults (shown in Section A.2 of the Appendix).

### 3.2 Preprocessing

For each screenshot $s_i$, we collected a participant's think-aloud verbal description $d_i$ together with the corresponding gaze data $G_i$. The verbal description $d_i$ was transcribed from audio using Turboscribe, which provides word-level timestamps. With visual supplementary

information from $s_i$, we use GPT-5.2 to segment $d_i$ into a sequence of $n_i$ utterances, $d_i \rightarrow \{d_{ij}\}_{j=1}^{n_i}$. Each segment $d_{ij}$ corresponds to a semantically-related UI element set $u_{ij}$ in the screenshot and has a summarized name $h_{ij}$ assigned by GPT-5.2. A semantically related UI set may include a single UI element (e.g., an icon) or a group of related elements (e.g., a card containing a title, image, and button, or a navigation bar). Segments without fixations or unrelated to UI contents are removed or re-segmented. To assess semantic coherence after cleaning, the authors of the paper audited 10% of the 7137 segments generated from participants' verbal descriptions; the segmentation coherence rate was 96.35%. The full audit protocol is described in Appendix A.5.

Each segment $d_{ij}$ is also associated with a time window with start time $t_{ij}^s$ and end time $t_{ij}^e$. We extract the corresponding subset of gaze data from $G_i$ that falls within the same time window, $G_{ij} = \{g \in G_i \mid t_{ij}^s \leq t(g) \leq t_{ij}^e\}$. In other words, the gaze behavior $G_{ij}$ is naturally coupled with the participant's verbal description $d_{ij}$ of UI set $u_{ij}$. It is noted that the coupling is not exact in every case: transitions, revisits, and browsing of related elements can produce small gaze–speech misalignments. A separate spatial grounding check on 200 elements reported in Appendix A.7.1 found direct overlap between the gaze cluster and verbalized element in 86.5% of cases, with an additional 11.0% falling on the target's immediate periphery and 2.5% misaligned. We retain these naturally occurring cases because they reflect the noise present in real-world gaze input rather than idealized gaze grounding. This design enables future systems to study robustness to such noise. More details about segmentation duration, segmentation word count, fixation duration, and fixation count per element distributions are shown in Figure A.1 in the Appendix.

## 4 Experiments

### 4.1 Gaze Visualizations

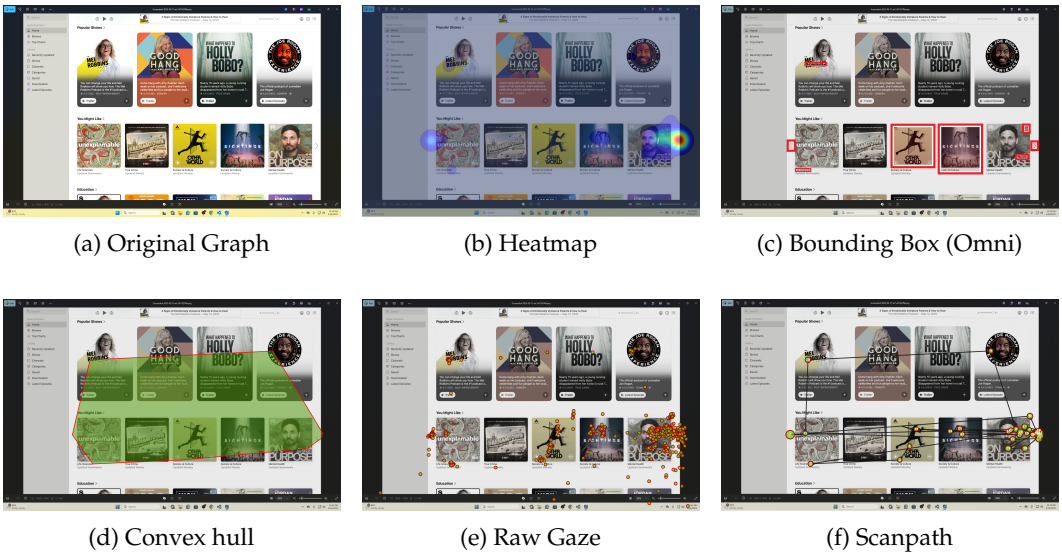
We visualize each gaze segment $G_{ij}$ using five methods:

$$\mathcal{R} = \{\text{heatmap},\ \text{convex hull},\ \text{scanpath},\ \text{bounding box (Omni)},\ \text{raw gaze}\}.$$

Applying a visualization $r \in \mathcal{R}$ produces an overlay on the original screenshot: $v_{ij}^{(r)} = f_r(G_{ij}, s_i)$. All visualizations follow standard practices in the literature. More specifically, raw gaze directly overlays sampled gaze points on the image, with duration reflected implicitly through repeated points and local cluster density. Fixation-based scanpaths represent the sequence and duration of fixations using numbered gaze circles whose size reflects viewing time. Convex hulls are the smallest polygons that enclose all fixation points. These three visualizations are rendered with red outlines and semi-transparent green fill, selected through iterative testing to maximize visibility while minimizing occlusion of underlying UI elements. For the remaining two visualizations: heatmaps convert gaze samples into smoothed spatial density maps, where warmer and more saturated colors indicate greater fixation density and duration; bounding Box (Omni) uses OmniParser (Lu et al., 2024) to detect UI elements from the screenshot and visualizes all detected bounding boxes containing one or more fixation points. In particular, Omni uses only a red outline (without green fill) to avoid implying the entire box is the gaze-covered area. Overall, scanpaths, heatmaps, and raw gaze preserve temporal information to varying degrees, whereas convex hulls and Omni emphasize spatial regions or gazed elements and discard temporal information.

### 4.2 Models

We evaluate five visualization methods across seven multimodal language models, resulting in 35 model–visualization conditions for each segmented UI element. We selected at least one small, medium, and large model from both open and proprietary families. The proprietary models are GPT-5-mini, Claude-Sonnet-4.6, GPT-5.2, and Gemini-3.1-Pro-Preview. The open models are Qwen3-VL-8B-Instruct, Llama-4-Maverick, and Qwen3-VL-235B-A22B-Instruct.



(a) Original Graph (b) Heatmap (c) Bounding Box (Omni)

(d) Convex hull (e) Raw Gaze (f) Scanpath

Figure 1: Five types of gaze visualizations. Participant's verbal description: "Similar idea, this is a carousel because there's another right arrow on the right side, meaning that there are more things that you're able to see towards right."

All primary model evaluations use zero-shot prompting; Appendix A.8.1 reports a small four-shot pilot using Gemini-3.1-Pro-Preview with heatmap and scanpath inputs.

### 4.3 Prediction Task

At prediction time, for each UI element set $u_{ij}$, we input only the original screenshot $s_i$ and a gaze visualization $v_{ij}^{(r)}$ (i.e., gaze visualization overlayed on the same screenshot) into a multimodal language model $M$; the model does not receive the participant's verbal description $d_{ij}$. The evaluated model produces a prediction of the attended UI element set, $\hat{u}_{ij}^{(M,r)} = M(s_i, v_{ij}^{(r)})$. The prediction is represented in text rather than as coordinates or a bounding box, including (1) a concise name for the element(s) and (2) a one-sentence description of what the element is and its location. Together, the two outputs specify the semantic identity and spatial scope of the prediction. We also collected visual, spatial, and functional properties for future research; for the present benchmark, we evaluated only (1) and (2). The full prompt template is provided in Appendix A.10.

### 4.4 Evaluation Metrics

Our data collection protocol naturally aligns gaze with participant-provided semantic ground truth: participants verbally describe the UI content they are attending to while their gaze is recorded. We therefore treat $d_{ij}$ as the ground truth for the participant's intended UI element set $u_{ij}$, without inferring intent from the gaze visualization itself. The evaluated model's task is therefore to recover these intended elements from the gaze visualization. Because both the ground truth and model prediction are expressed in text, a direct semantic-similarity comparison would be insufficient: for example, a participant may describe a button's color while the model describes the same button using other properties. We therefore adopt a coverage-based evaluation scheme that focuses on whether the LLM correctly identifies the UI region corresponding to the element(s) described by the participant.

Operationally, we ground the semantic ground truth $d_{ij}$ to its corresponding target region in the original screenshot, and similarly ground the evaluated model's predicted element name and description to a predicted region.Through manual inspection of LLM responses,

we developed a five-category coverage taxonomy: **Perfect Coverage**—the predicted region matches the target at comparable granularity (including immediate parent containers and visual property /element equivalences); **Partial Coverage**—the prediction is a subset of the target (e.g., one of several described elements of sub-components); **Overcoverage**—the prediction includes unrelated areas beyond the target; **Partial + Overcoverage**—the prediction covers a subset while also introducing irrelevant elements; and **Miss**—no meaningful overlap between predicted and target regions. Appendix A.6 provides a worked example showing how the same target description is labeled under all five prediction outcomes.

To apply this taxonomy at scale, we use GPT-5.2 as an LLM judge; judge calls are separate from all evaluated-model calls. The judge receives the participant's verbal description $d_{ij}$, a surrounding verbal context, the evaluated model's predicted element name and element description, and the original screenshot $s_i$ without the gaze overlay. It first grounds the target element(s) described by $d_{ij}$ in the screenshot, then grounds the evaluated model's textual prediction, and finally compares their overlap, containment, and granularity. The surrounding verbal context—a window of five segments before and after, $d_{ij-5} \ldots d_{ij+5}$—and the original screenshot are used only to disambiguate references. Importantly, the judge never receives the gaze visualization: gaze is the evaluated model's input, whereas the participant's verbal description provides the semantic ground truth used for judging. Verbal descriptions often include appearance, position, function, or surrounding context, providing enough context to identify the intended element among same-type alternatives.

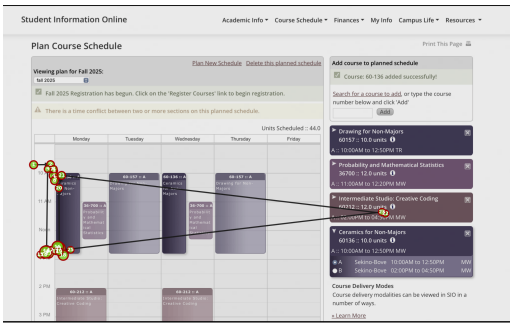
To validate the coverage labels and the LLM judge, three trained annotators, who are the authors of this paper, audited the results of 600 UI elements, which are sampled randomly and non-overlappingly across all models. Each annotator labeled all five visualization-conditioned predictions for their assigned elements, yielding 3,000 human judgments. The human annotators received the same inputs as the LLM judge and followed the same comparison procedure; they did not inspect or interpret the gaze visualizations when assigning labels. The complete training, annotation, and reliability protocol is provided in Appendix A.5. Overall human–LLM agreement is 81.2%. Over half of the disagreements (11.1% of all labels) involve cases where human annotators assigned Perfect Coverage but the LLM judge did not, typically when the prediction references the containing section or partial visual properties of the target. This suggests the LLM judge applies a more conservative standard. Given that the disagreements reflect differences in leniency rather than fundamental labeling errors, we use LLM-as-a-judge to complete the largely scaled annotation for all 7,137 elements across 5 visualization conditions and 7 models. Because GPT-5.2 is both an evaluated prediction model and the primary LLM judge, we reran all 35 model–visualization conditions on a random 200-element subset using Grok-4.3, which is not among the evaluated models. The two judges agreed on 80.2% of 7,000 labels and produced nearly identical model rankings, as reported in Appendix A.7.2.

## 5 Results

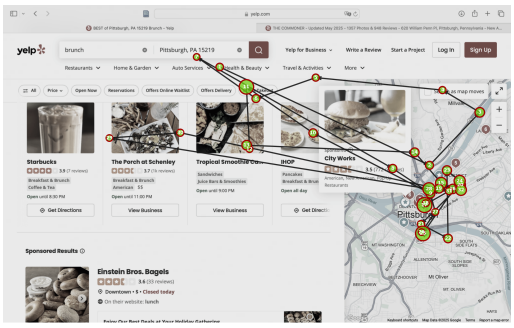
### 5.1 Overall Performance

Across all seven models and five visualizations, the coverage label distribution is strongly bimodal: miss is the most frequent label at approximately 40.0%, followed by overcoverage at 33.8%, together accounting for nearly 74% of all judgments. Perfect coverage contributes approximately 14.0%, partial coverage 9.5%, and partial + overcoverage only 2.7%. The dominance of miss and overcoverage indicates that current models tend to either fail to identify the attended element entirely or predict a region substantially larger than the user's actual focus.

Across visualizations, Gemini-3.1-Pro-Preview achieves the highest perfect coverage under four of the five visualizations (boldface in Table 1). GPT-5.2 achieves the lowest average miss rate of any model (31.3%). Sonnet-4.6 and GPT-5.2 are the two runners-up in average perfect coverage, achieving 14.7% and 14.4%, respectively. Among open-weight models, Qwen3-VL-235B-A22B-Instruct generally outperforms Llama-4-Maverick and Qwen3-VL-




(a) A scanpath visualization on a course scheduling platform. Participant's verbal description: "For example the ceramics class begins at 10 a.m, and you can see that it extends almost up to 1 p.m, but not entirely, so you know that it's going to end a little before 1 p.m."




(b) A scanpath visualization on Yelp. Participant's verbal description: "And just briefly connect that with what you're seeing under the header, there's a map which the user should be able to navigate. And so, in other words, they can also refine their results visually."

Figure 2: Gaze segments with the participant's concurrent verbal description, shown as scanpaths. Original screenshots are in Appendix A.6 (Figures A.4 and A.5).

8B-Instruct. Llama-4-Maverick exhibits the highest miss rate across nearly all conditions, suggesting limited capacity to reason about gaze visualizations.

### 5.2 Effect of Gaze Visualization

Treating the judge outcomes as nominal categories (*miss*, *partial coverage*, *partial + overcoverage*, *overcoverage*, and *perfect coverage*), we found a strong overall effect of visualization on the label distribution. A cluster-preserving permutation omnibus test on the pooled $5 \times 5$ visualization-by-label table was significant ($\chi^2(16) = 6302.09$, $p = 0.0005$), confirming that the five visualizations produce distinct coverage patterns overall. All pairwise full-categorical comparisons were also significant after Holm correction.

The resulting ranking is: heatmap $>$ scanpath $>$ raw gaze $>$ convex hull $\approx$ omni. Heatmap yields the fewest miss labels (36.1%) and the highest perfect coverage (19.8%) and partial coverage (12.9%). Label-specific follow-up tests confirmed that heatmap had significantly fewer misses and more perfect coverage than every alternative (all Holm-adjusted $p = 0.010$). Heatmaps succeed because aggregating gaze into continuous density surfaces naturally aligns with how vision models process spatial and attention information, while also filtering noise from incidental fixations with single or minimal focus—yielding the lowest overcoverage rate (28.4%).

Scanpath achieves the second-best performance (14.8% perfect coverage, 36.8% miss) but produces the highest overcoverage rate (38.0%): point-based visualizations present every fixation as a discrete marker, making it difficult for models to distinguish primary attention from incidental glances. The substantially lower perfect coverage and higher miss rates observed for convex hull, omni, and raw gaze show that spatial data alone, stripped of temporal duration, is insufficient—no matter whether the spatial signal is compressed into a single encompassing region or spread across all recorded points.

### 5.3 Effect of UI Element Type

We categorized UI elements into 21 categories based on a list adapted from prior work (Leiva et al., 2020; Peng et al., 2024); exact definitions are provided in the Appendix A.4. Figure A.3 in Appendix A.4 shows the full list of perfect coverage rate for each UI type–visualization pair, sorted by mean perfect coverage across visualizations. The overall visualization ranking is largely stable across UI types, confirming that visualization choice matters more as a global design decision than as a per-element one. However, the magnitude of each visualization's advantage varies systematically with element type.

UI element type is itself a strong predictor of benchmark performance, and this ordering is consistent across all five visualizations. Structured, spatially extended elements—Table (mean perfect coverage 32.9%), List (21.6%), Grid (19.6%), and Chart (19.7%)— are the high ranking elements. Varying, contained elements including Icon (9.5%), Dropdown (13.9%), and Button (12.9%) are among the lowest ranked elements. To understand this gap, we examined the underlying gaze patterns in our dataset and found that UI types are associated with two attentional modes. The high ranking elements tend to elicit many short fixations with longer scanpaths and spatially diffuse gaze. The low ranking elements tend to elicit long fixation durations but few fixations, with spatially compact gaze clouds. Fixation duration and fixation count are nearly uncorrelated across elements ($r = -0.025$), confirming these are separable dimensions rather than two ends of one continuum.

This two-diagonal attentional mode helps explain why the visualization ranking shifts. For compact elements with tight clusters and sustained attention, heatmap's spatial smoothing produces a focused, unambiguous hotspot. Heatmap's advantage is most pronounced in exactly these categories: Dropdown (22.9% vs. next-best 12.0%), and retains its advantage even for Switch/Checkbox (21.9% vs. 14.1%), where fixation counts and durations are both high due to gaze revisits. For spatially extended elements scanpath matches or outperforms heatmap (Table: 44.1% vs. 35.7%; Grid: 24.3% vs. 19.3%). For example, grid, which includes captures carousels, card collections, and multi-row layouts where users employ Z-shaped traversal patterns arocss rows (de Leon-Martinez et al., 2025), scanpath preserves sequential structure —allowing models to infer that separated fixation clusters rather than smoothing them into a single, ambiguous warm region.

Two categories remain difficult have unique challenges: Background (7.6%) lacks clear visual boundaries, and Text (11.5%) frequently co-occurs with functional containers, causing models to attribute gaze to surrounding structures rather than the text itself.

## 6 Discussion

### 6.1 How could LLMs do better?

The significant overcoverage rates observed across conditions reveal three distinct failure modes in how LLMs interpret gaze data.

**Failure Mode 1: Inability to parse dense interfaces.** In dense layouts where many elements are tightly packed, models struggle to isolate individual components, leading to spatially imprecise predictions that bleed into neighboring regions.

**Failure Mode 2: Conflating spatial extent with attentional scope.** When fixations are scattered across the screen, models treat each fixated region independently rather than recognizing that multiple gaze points may serve a single underlying goal. The model lacks the ability to group dispersed fixations into a functionally coherent unit, reasoning about *where the user looked* rather than *what the user was trying to do*.

**Failure Mode 3: Inability to distinguish signal from noise in gaze.** LLMs treat nearly every marked gaze point as meaningful, lacking the ability to filter noise inherent in gaze data. This is most evident in overcoverage rates for raw gaze and scanpath visualizations. In practice, gaze is noisy and frequently misaligned with the true target—fixations drift toward salient periphery elements or land on contextually related but non-target components. Without the capacity to discount incidental fixations, models inflate predicted regions well beyond the user's actual focus.

These failure modes are not equally distributed across models. Through inspection of LLM reasoning, we found that Gemini-3.1-Pro-Preview, the top performer, stays faithfully to the visual information and adapts prediction to the granularity of each visualization with light but meaningful inference to user intentionality and functional unity of viewed regions. In contrast, Llama-4-Maverick frequently states the user's focus without referencing the gaze visualization—consistent with its high miss rates. Over interpretation of user intentionality or gaze information potentially lead to low performance in Qwen families. This suggests

| Visual Type | Coverage | GPT-5.2 * | Gemini-3.1 | Sonnet-4.6 | GPT-5-mini | Qwen3-235B | Llama-4-Mav | Qwen3-8B |
|---|---|---|---|---|---|---|---|---|
| **Convex Hull** | PC | 12.3 | **14.3** | 11.5 | 11.6 | 12.6 | 10.8 | 11.6 |
| | M | 36.7 | 41.3 | 40.4 | 37.3 | 43.4 | 55.4 | 47.9 |
| | OC | 41.5 | 34.5 | 37.5 | 40.9 | 35.7 | 24.1 | 32.7 |
| | PaC | 8.0 | 8.4 | 8.4 | 7.7 | 7.8 | 9.2 | 7.3 |
| | PaOC | 1.6 | 1.5 | 2.2 | 2.6 | 0.6 | 0.6 | 0.5 |
| **Heatmap** | PC | 20.8 | **27.0** | 21.0 | 15.1 | 19.7 | 16.4 | 18.3 |
| | M | 28.5 | 37.3 | 36.2 | 27.8 | 35.0 | 49.8 | 38.2 |
| | OC | 32.2 | 18.2 | 24.1 | 37.3 | 33.4 | 21.1 | 32.2 |
| | PaC | 13.1 | 16.7 | 16.5 | 11.0 | 10.5 | 12.0 | 10.6 |
| | PaOC | 5.4 | 0.8 | 2.2 | 8.7 | 1.4 | 0.6 | 0.6 |
| **Omni** | PC | 9.0 | 10.5 | 10.0 | 9.1 | 11.9 | 10.4 | **14.0** |
| | M | 33.5 | 35.0 | 39.2 | 34.1 | 40.4 | 61.0 | 49.9 |
| | OC | 36.7 | 41.0 | 33.6 | 35.8 | 35.2 | 16.8 | 25.4 |
| | PaC | 9.2 | 9.0 | 10.5 | 8.8 | 9.4 | 10.9 | 10.2 |
| | PaOC | 11.6 | 4.5 | 6.7 | 12.2 | 3.1 | 0.8 | 0.6 |
| **Raw Gaze** | PC | 13.0 | **20.5** | 15.0 | 10.4 | 11.0 | 9.0 | 10.5 |
| | M | 29.5 | 32.7 | 33.6 | 33.5 | 50.9 | 60.1 | 52.6 |
| | OC | 47.3 | 36.4 | 40.7 | 42.2 | 30.2 | 23.1 | 29.2 |
| | PaC | 7.9 | 9.6 | 9.5 | 8.4 | 7.2 | 7.3 | 7.3 |
| | PaOC | 2.3 | 0.8 | 1.2 | 5.5 | 0.7 | 0.5 | 0.5 |
| **Scanpath** | PC | 16.7 | **23.9** | 16.1 | 11.9 | 12.4 | 11.4 | 10.9 |
| | M | 28.1 | 32.3 | 31.3 | 29.9 | 40.4 | 48.0 | 47.8 |
| | OC | 43.9 | 30.7 | 41.5 | 43.7 | 39.4 | 32.9 | 33.6 |
| | PaC | 9.0 | 12.4 | 9.1 | 8.2 | 6.9 | 7.1 | 7.2 |
| | PaOC | 2.3 | 0.9 | 1.9 | 6.2 | 0.8 | 0.6 | 0.5 |

Table 1: Coverage distribution (%) by visualization and model. PC = perfect coverage, M = miss, OC = overcoverage, PaC = partial coverage, PaOC = partial + overcoverage. Bold marks the highest PC per visualization. *GPT-5.2 also serves as the primary LLM judge; evaluation and judging calls are separate.

that prompting models to reason explicitly about gaze properties (density, spread, temporal order) and underlying user intentionality may improve focus prediction.

### 6.2 What Gaze Reveals Beyond Coverage

The examples in our dataset illustrate that gaze carries far richer information than spatial coverage alone. Across diverse tasks and interfaces, we observe gaze behaviors that encode distinct forms of user intentionality—most of which current models fail to capture.

In Figure 2a, the scanpath reveals two fixation clusters—one near "10am" and another near "1pm"—connected by a saccade, while the salient class title receives little attention. The subsequent shift to a corresponding class card suggests the user is verifying start and end times. This demonstrates **inferring intent from selective attention** and **revealing relationships between spatially distant elements**—capabilities that coverage alone cannot capture. In Figure 2b, long scanpaths with varying fixation durations indicate **visual search and cross-referencing** between listed items and map positions. In Figure 1f, the user's gaze traverses two UI sections quickly but dwells noticeably longer on a right-arrow navigation element, showing how **fixation duration differentiates browsing from intentional focus**. These temporal patterns—rapid saccades, comparison sequences, dwell contrasts—are precisely the signals that models miss when they reduce gaze to a static spatial region.

These case studies also highlight when scanpaths may be more informative than heatmaps: attentional shifts are often instantaneous, and scanpaths preserve the temporal precision of such transitions, whereas heatmaps smooth them out.

## 7 Limitations and Future Work

All of our participants had at least some UI/UX design experience, but only four reported professional experience. This experience supported fluent verbalization and gaze–speech alignment, but the resulting data may not generalize to less experienced users. Future studies could recruit a larger participant pool with varying levels of UI/UX experience. In addition, GPT-5.2 is both one of the evaluated models and the LLM judge. Although the evaluation and judging calls are separate, this dual role may introduce self-judge bias. In an audit of 200 sampled instances, an independent Grok-4.3 judge agreed with GPT-5.2 on 80.2% of labels and produced a similar model ranking. Still, the audit is not on the full set and the two judges disagreed on a fifth of the labels, so some self-judge bias may remain.

Future benchmarks should cover dynamic and interactive UIs, including click and hover states, navigation across linked webpages, and multi-step interaction sequences. Future work should also explicitly model higher-level, sequential gaze dynamics such as fixation transitions, revisits, temporal ordering, and task-level intent over time. The benchmark could be enriched with multidimensional behavioral annotations that capture user intent and goals, task context, and action outcomes, which could ultimately serve as implicit signals for human-centered UI assessment (Wu et al., 2024; Peng et al., 2025a; 2026b) and automation (Yang et al., 2026; Peng et al., 2025b). Finally, our four-shot pilot improved results only marginally, so a handful of in-context examples is not enough. Prompting strategies tailored to each visualization and lightweight fine-tuning are the next steps for getting models to better interpret and use gaze.

## 8 Conclusion

We introduce GLAZE, the first gaze-language grounding benchmark for web UIs, which pairs screenshots and gaze traces with the concurrent verbal descriptions that label what each participant attended to. Across seven MLLMs and five gaze visualizations, models usually either miss the attended element or predict a region far larger than the user's focus. Of the five, heatmaps help models most, but even then models are fully correct less than one-fifth of the time. Systems that act on gaze today therefore risk targeting the wrong element. We hope this benchmark supports future work that brings gaze into multimodal reasoning, UI agents, and automated interface assessment.

## Generative AI Use Statement

We used an LLM-based tool for language polishing; all content and claims were authored and verified by the authors.

## References

Julio Abascal, Myriam Arrue, and Xabier Valencia. Tools for web accessibility evaluation. In *Web accessibility: a foundation for research*, pp. 479–503. Springer, 2019.

Miguel Carrasco, César González-Martín, José Aranda, and Luis Oliveros. Vision transformer attention alignment with human visual perception in aesthetic object evaluation. *PLoS One*, 21(4), April 2026. doi: 10.1371/journal.pone.0344006.

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

Santiago de Leon-Martinez, Jingwei Kang, Robert Moro, Maarten de Rijke, Branislav Kveton, Harrie Oosterhuis, and Maria Bielikova. Recgaze: the first eye tracking and user interaction dataset for carousel interfaces. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3702–3711, 2025.

Soussan Djamasbi, Marisa Siegel, and Tom Tullis. Visual hierarchy and viewing behavior: An eye tracking study. In Julie A. Jacko (ed.), *Human-Computer Interaction. Design and Development Approaches*, pp. 331–340, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-21602-2. doi: 10.1007/978-3-642-21602-2_36.

Ramy ElMallah, Mohamed Abubakr Hassan, Nima Zamani, and Chi-Guhn Lee. Gaze2Instruct (G2I): Towards a more inclusive language-conditioned robotic assistance for severe speech and motor impairments. *International Journal of Human–Computer Interaction*, 42(3):1892–1908, 2026.

Anna Maria Feit, Shane Williams, Arturo Toledo, Ann Paradiso, Harish S. Kulkarni, Shaun K. Kane, and Meredith Ringel Morris. Toward everyday gaze input: Accuracy and precision of eye tracking and implications for design. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 1118–1130, New York, NY, USA, 2017. Association for Computing Machinery. doi: 10.1145/3025453.3025599.

Karim Galliamov, Ivan Titov, and Ilya Pershin. Enhancing RLHF with human gaze modeling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 30625–30631, 2025.

Lina Gao, Changyuan Wang, and Gongpu Wu. Wearable biosensor smart glasses based on augmented reality and eye tracking. *Sensors*, 24(20):6740, 2024. doi: 10.3390/s24206740.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14281–14290. IEEE, 2024.

Yue Jiang, Luis A. Leiva, Hamed Rezazadegan Tavakoli, Paul R. B. Houssel, Julia Kylmälä, and Antti Oulasvirta. UEyes: Understanding visual saliency across user interface types. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394215. doi: 10.1145/3544548.3581096. URL https://doi.org/10.1145/3544548.3581096.

Robert Konrad, Nitish Padmanaban, J Gabriel Buckmaster, Kevin C Boyle, and Gordon Wetzstein. Gazegpt: Augmenting human capabilities using gaze-contingent contextual ai for smart eyewear. *arXiv preprint arXiv:2401.17217*, 2024.

Ricardo Bigolin Lanfredi, Mingyuan Zhang, William F. Auffermann, Jessica Chan, Phuong-Anh T. Duong, Vivek Srikumar, Trafton Drew, Joyce D. Schroeder, and Tolga Tasdizen. Reflacx, a dataset of reports and eye-tracking data for localization of abnormalities in chest x-rays. *Scientific Data*, 9(1):350, 2022. doi: 10.1038/s41597-022-01441-z.

Daeun Lee, Subhojyoti Mukherjee, Branislav Kveton, Ryan A Rossi, Viet Dac Lai, Seunghyun Yoon, Trung Bui, Franck Dernoncourt, and Mohit Bansal. Streamgaze: Gaze-guided temporal reasoning and proactive understanding in streaming videos. *arXiv preprint arXiv:2512.01707*, 2025.

Jungbeom Lee, Sanghyuk Chun, and Sangdoo Yun. Toward interactive regional understanding in vision-large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6416–6429, Mexico City, Mexico, jun 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.356. URL https://aclanthology.org/2024.naacl-long.356/.

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

Yiwei Li, Zihao Wu, Yanjun Lv, Hanqi Jiang, Weihang You, Zhengliang Liu, Dajiang Zhu, Xiang Li, Quanzheng Li, Tianming Liu, et al. Thinking with gaze: Sequential eye-tracking as visual reasoning supervision for medical vlms. *arXiv preprint arXiv:2603.06697*, 2026.

Weifeng Lin, Xinyu Wei, Ruichuan An, Peng Gao, Bocheng Zou, Yulin Luo, Siyuan Huang, Shanghang Zhang, and Hongsheng Li. Draw-and-understand: Leveraging visual prompts to enable MLLMs to comprehend what you want. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=bfa58H1nQ8.

Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. OmniParser for pure vision based GUI agent, 2024. URL https://arxiv.org/abs/2408.00203.

Michele Mazzamuto, Antonino Furnari, Yoichi Sato, and Giovanni Maria Farinella. Gazing into missteps: Leveraging eye-gaze for unsupervised mistake detection in egocentric videos of skilled human activities. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8310–8320, 2025.

Sounak Mondal, Naveen Sendhilnathan, Ting Zhang, Yue Liu, Michael Proulx, Michael Louis Iuzzolino, Chuan Qin, and Tanya R. Jonker. Gaze-language alignment for zero-shot prediction of visual search targets from human gaze scanpaths. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2738–2749, October 2025.

Peya Mowar, Yi-Hao Peng, Jason Wu, Aaron Steinfeld, and Jeffrey P Bigham. Codea11y: Making ai coding assistants useful for accessible web development. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2025.

Débora Maria Barroso Paiva, André Pimenta Freire, and Renata Pontin de Mattos Fortes. Accessibility and software engineering processes: A systematic literature review. *Journal of Systems and Software*, 171:110819, 2021.

Anupam Pani and Yanchao Yang. Gaze-VLM: Bridging gaze and VLMs through attention regularization for egocentric understanding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://neurips.cc/virtual/2025/loc/san-diego/poster/120280.

Alexandra Papoutsaki, James Laskey, and Jeff Huang. Searchgazer: Webcam eye tracking for remote studies of web search. In *Proceedings of the 2017 conference on conference human information interaction and retrieval*, pp. 17–26, 2017.

Taiying Peng, Jiacheng Hua, Miao Liu, and Feng Lu. In the eye of MLLM: Benchmarking egocentric video intent understanding with gaze-guided prompting. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026a. URL https://neurips.cc/virtual/2025/loc/san-diego/poster/121584.

Yi-Hao Peng, Peggy Chi, Anjuli Kannan, Meredith Ringel Morris, and Irfan Essa. Slide gestalt: Automatic structure extraction in slide decks for non-visual access. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2023.

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

Yi-Hao Peng, Jeffrey P Bigham, and Jason Wu. Designpref: Capturing personal preferences in visual design generation. *arXiv preprint arXiv:2511.20513*, 2025a.

Yi-Hao Peng, Dingzeyu Li, Jeffrey P Bigham, and Amy Pavel. Morae: Proactively pausing ui agents for user choices. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–14, 2025b.

Yi-Hao Peng, Samarth Das, Jeffrey P. Bigham, and Jason Wu. Efficient personalization of generative user interfaces. *arXiv preprint arXiv:2604.09876*, 2026b.

Alexander Plopski, Teresa Hirzle, Nahal Norouzi, Long Qian, Gerd Bruder, and Tobias Langlotz. The eye in extended reality: A survey on gaze interaction and eye tracking in head-worn extended reality. *ACM Computing Surveys*, 55(3):1–39, 2023. doi: 10.1145/3491207.

Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *European conference on computer vision*, pp. 647–664. Springer, 2020.

Gabriel Herbert Sarch, Balasaravanan Thoravi Kumaravel, Sahithya Ravi, Vibhav Vineet, and Andrew D Wilson. Grounding task assistance with multimodal cues from a single demonstration. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 12807–12833, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.663. URL https://aclanthology.org/2025.findings-acl.663/.

Maarten W Van Someren, Yvonne F Barnard, Jacobijn AC Sandberg, et al. *The think aloud method: a practical guide to modelling cognitive processes*. Academic Press, London, UK, 1994. ISBN 0-12-714270-3.

Paul Voigtlaender, Soravit Changpinyo, Jordi Pont-Tuset, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with video localized narratives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2461–2471, June 2023.

Jason Wu, Siyan Wang, Siman Shen, Yi-Hao Peng, Jeffrey Nichols, and Jeffrey P Bigham. Webui: A dataset for enhancing visual ui understanding with web semantics. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2023.

Jason Wu, Yi-Hao Peng, Xin Yue Amanda Li, Amanda Swearngin, Jeffrey P. Bigham, and Jeffrey Nichols. UIClip: A data-driven model for assessing user interface design. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24)*. Association for Computing Machinery, 2024. doi: 10.1145/3654777.3676408.

Qianhui Wu, Kanzhi Cheng, Rui Yang, Chaoyun Zhang, Jianwei Yang, Huiqiang Jiang, Jian Mu, Baolin Peng, Bo Qiao, Reuben Tan, et al. Gui-actor: Coordinate-free visual grounding for gui agents. *Advances in Neural Information Processing Systems*, 38:15101–15128, 2026.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, et al. Os-atlas: Foundation action model for generalist gui agents. In *International Conference on Learning Representations*, volume 2025, pp. 5090–5108, 2025.

Kun Yan, Zeyu Wang, Lei Ji, Yuntao Wang, Nan Duan, and Shuai Ma. Voila-a: Aligning vision-language models with user's gaze attention. *Advances in neural information processing systems*, 37:1890–1918, 2024.

Saelyne Yang, Jaesang Yu, Yi-Hao Peng, Kevin Qinghong Lin, Jae Won Cho, Yale Song, and Juho Kim. Guide: A benchmark for understanding and assisting users in open-ended gui tasks. *arXiv preprint arXiv:2603.25864*, 2026.

Yifan Zhang, Jiliang Li, Zachary Karas, Aakash Bansal, Toby Jia-Jun Li, Collin McMillan, Kevin Leach, and Yu Huang. Eyetrans: Merging human and machine attention for neural code summarization. *Proceedings of the ACM on Software Engineering*, 1(FSE):115–136, 2024.

Zejia Zhang, Bo Yang, Xinxing Chen, Weizhuang Shi, Haoyuan Wang, Wei Luo, and Jian Huang. MindEye-OmniAssist: A gaze-driven LLM-enhanced assistive robot system for implicit intention recognition and task execution. In *2025 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, pp. 1–6. IEEE, 2025.

# A Appendix

All analysis below, if not specified, focuses on the 7137 elements created from user data. Appendix §A.3 and Appendix §A.4 provide more details about the 3280 elements created by the experimenters.

## A.1 Participant Demographics

Seventeen participants identified themselves as females and ten participants identified themselves as males. Participants ranged in age from 18 to 44 years: 16 were aged 18–24, 10 were aged 25–34, and one was aged 35–44. As shown in Table A.1 and Table A.2, on a six scale experience sheet from no limited, moderate, professional, participant self-reported diverging experience. Only 4 people indicated that they have professional experience. The rest of the participants had diverse experience levels, with roughly even representation from limited to moderate experience. Eleven people have several weeks of experience and 9 said they have experience over a year.

| UI/UX Design Experience Level | Number of Participants |
|---|---|
| Limited experience (e.g., created low-fidelity prototypes or wire-frames) | 8 |
| Between limited and moderate experience (e.g., conducted early-stage user testing or created mid-fidelity prototypes) | 7 |
| Moderate experience (e.g., contributed to a project, conducted iterative design with users or software developers, and/or created high-fidelity prototypes) | 8 |
| Professional experience (e.g., worked as a professional UI/UX de-signer) | 4 |

Table A.1: Participants' prior experience with UI/UX design.

| Duration of UI/UX Design Experience | Number of Participants |
|---|---|
| Several weeks | 11 |
| Several months | 6 |
| Up to one year | 1 |
| Over one year | 9 |

Table A.2: Duration of participants' prior UI/UX design experience.

## A.2 Eye-Tracking Apparatus and Configuration

Gaze data were recorded on a display with an active area of $517.78 \times 323.62$ mm. Fixations were identified using a velocity-based algorithm with the following parameters: a velocity threshold of 30 degrees per second, a minimum fixation duration of 60 ms, a maximum intersample gap of 75 ms, and a merge angle of 0.5 degrees.

## A.3 Segmentation Statistics

Figure A.1 presents histograms of six per-element statistics for the participant-collected dataset. The distributions are strongly right-skewed with heavy tails, indicating that a small fraction of elements attracted disproportionately extended engagement, while most received relatively brief attention. We further compared the participant- and experimenter-collected subsets using five aggregate behavioral measures. For participants and the experimenter, respectively, mean segment duration was 8.0s versus 10.1s, mean word count was 20.3 versus 21.3, mean fixation count was 17.9 versus 23.3, mean fixation duration was 0.242s versus 0.220s, mean fixation spatial variability was 0.152 versus 0.148, and mean scanpath

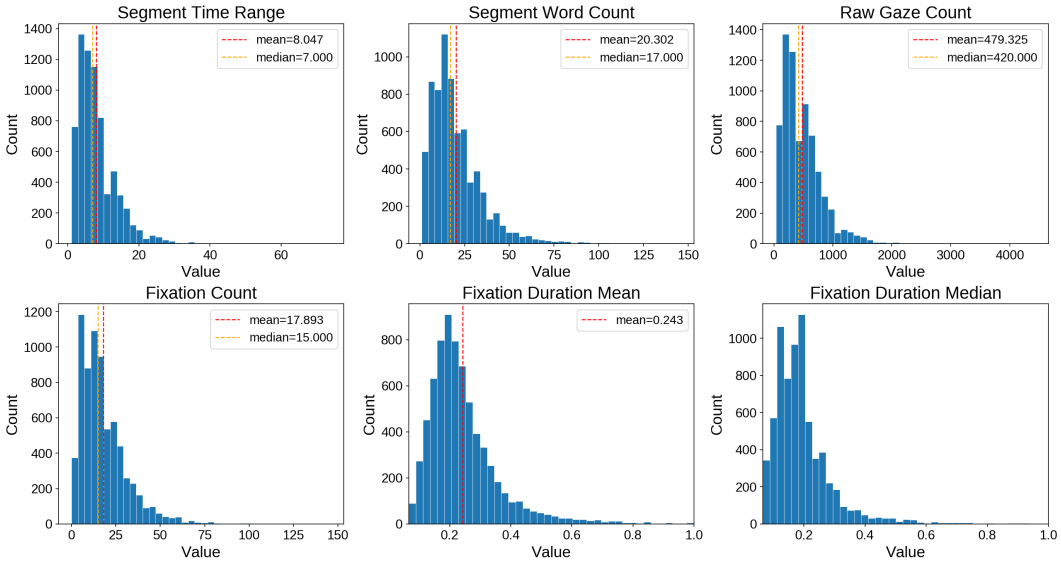



Figure A.1: Histograms of six per-element statistics across all categorized elements.

length was 2.107 versus 2.682 normalized screen units. When treating each participant as a separate data contributor, the experimenter's value was neither the highest nor the lowest for any measure. The experimenter-collected data are broadly consistent with the participant-collected data on these aggregate measures, although differences in other aspects of gaze behavior may remain.

### A.4 Screenshot and UI category Breakdown

Some screenshots come from the same website but depict substantially different pages, layouts, or content, enabling future researchers to examine within-site variation. Most screenshots come from unique websites. The only repeated websites account for 40 screenshots from 18 websites, with each website represented by 2–3 distinct screenshots.

To ensure consistent annotation, each UI element in the dataset was assigned to one of 21 categories.

1. **Icon.** A glyph used to represent something else. This includes logos, brand logos, ratings/stars, and small marks or overlays such as view counts or video duration shown on a thumbnail.
2. **Text.** A standalone piece of text, such as a description or paragraph. This also includes labels, tags, badges, and textual overlays such as views or duration.
3. **Header.** A fixed region at the top of the screen containing a title, subtitle, or other identifying label.
4. **Button.** A clickable call-to-action element expressed through text, an icon, or both. Examples include hamburger menus, customer help, settings, user profile, sign in, sign up, register, play, pause, close, cancel, thumbs up/down, long horizontal action bars, text buttons with icons, and hyperlinks.
5. **Image.** A standalone image, including a video thumbnail, cover image, profile image, or avatar.
6. **Card.** A UI container that groups content about a single subject, often combining text and images, such as a shopping item or a single video.
7. **Grid.** A collection of multiple cards arranged in one row or multiple rows. This category also includes carousels and card collections.

8. **Popup.** A callout, notification, reminder, alert, tooltip, or explanatory panel, including advertisement panels.
9. **Tab / Segmented Control.** A group of buttons or icons used to switch between views or states, where the selected option is visually highlighted.
10. **Slider.** A range selector such as a volume slider or price slider. This category also includes scroll bars, progress bars, timelines, and status bars.
11. **Date-picker.** A control used to select a date, such as a calendar widget on a website.
12. **Section / Container / Panel / Banner.** A bounded UI region that groups related content or functionality, such as a cookie banner, advertisement banner, left sidebar section, top navigation section, or main content panel.
13. **Search Bar / Text Field.** An input element used for entering text, including search bars, text fields, and text input boxes.
14. **List.** A vertically stacked set of items, including checked lists and simple stacked bars or entries.
15. **Navigation Bar.** A menu bar or toolbar used for navigation.
16. **Map / Canvas.** An interactive map, visualization, or canvas-based interface.
17. **Dropdown.** An expandable menu or selector, often attached to a navigation bar or input field, that opens downward to reveal options.
18. **Table.** A row-and-column structure with headers, where each cell typically contains simple text. Unlike a grid, a table is not primarily composed of cards.
19. **Background.** A background image, decorative background region, or overlay layer behind the main content.
20. **Chart / Graph / Diagram.** A visual representation of data or structure, such as a line chart, bar chart, pie chart, or interactive diagram, often including axes, labels, titles, or descriptions.
21. **Switch / Checkbox.** A toggleable control representing a binary state, including switches, toggles, and checkbox-like buttons.

Figure A.2 reports the frequency of each category in the annotated dataset. Text and Button together account for nearly half of all elements, whereas Chart, Switch/Checkbox, Background, and Slider each contain fewer than 100 instances. Note that one element could satisfy one or multiple categories at the same time. That is why the total count might exceed 7137 or 3280.

Figure A.3 reports the Perfect Coverage rate for each UI type under the five gaze visualizations, averaged across all evaluated models. Rows are sorted by mean Perfect Coverage rate, and the outlined cell marks the best-performing visualization for each UI type. Heatmap and scanpath account for nearly all per-type best performances.

### A.5 Annotation Validation Protocol

**Validation of gaze–language segments.** Our think-aloud protocol produces tightly coupled gaze and speech: changing a segmentation boundary changes both modalities together and can therefore yield a different but equally valid gaze–language pair. We recruited participants with UI/UX experience who could verbalize screen content fluently, reducing potential gaze–speech misalignment. We audited whether each resulting pair was semantically coherent, meaning that the gaze window and verbal segment referred to the same UI element set. An example of a failure case is a segment in which the gaze covers a title and button, while the verbal segment contains a complete reference to the title and only a partial clause about the button.

We randomly sampled 1,042 segments, corresponding to 10% of the dataset. The segments were divided among three annotators, who are the authors of the paper, with each segment reviewed by one annotator. After the initial audit, one annotator revisited all segments marked incoherent using the source video and full transcript to verify the judgment. The resulting segment-coherence rate was 96.35%.

**Validation of coverage labels.** Three trained annotators audited 600 base instances sampled across models. The instances were divided approximately evenly, with each initially re-

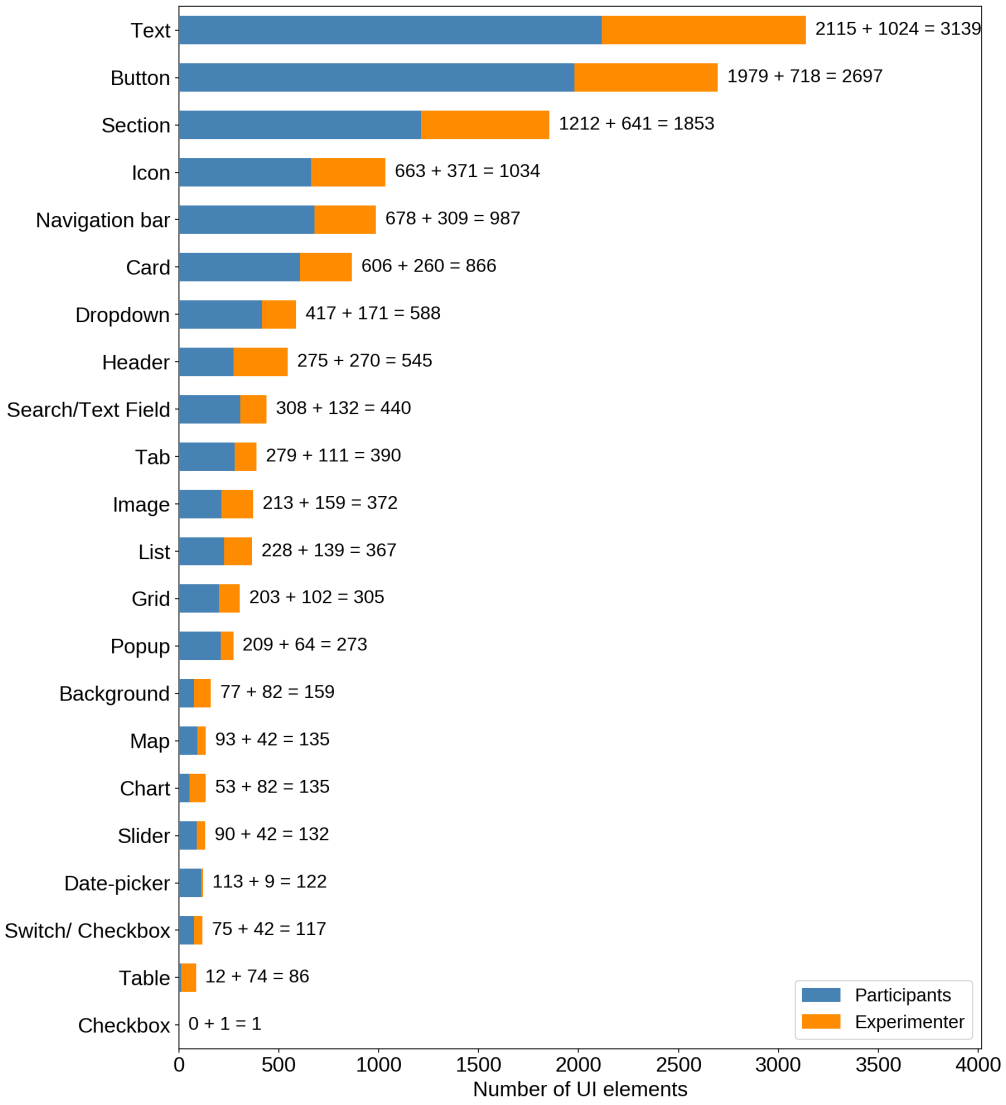



Figure A.2: Element count per category from the 7137 elements based on participant data and the 3280 elements based on experimenter data.

viewed by one annotator. For each instance, the annotator labeled all five gaze-visualization conditions, yielding 3,000 human coverage judgments. Annotators followed a shared written codebook containing the five coverage definitions and decision rules; the LLM-judge prompt was derived from this codebook. Before annotation, all three annotators jointly labeled held-out examples and discussed their decisions until no questions about the criteria remained.

To measure inter-annotator reliability, all three annotators independently labeled a randomly selected overlap subset of 72 base instances across five visualizations, yielding 360 shared judgments per annotator. Pairwise exact agreement was A/B: 91.1%, B/C: 91.9%, A/C: 95.3%. Milestone meetings were held to discuss edge cases, align expectations, and correct inconsistent applications of the criteria. The resulting human labels were used to validate the GPT-5.2 judge, which achieved 81.2% exact agreement with the human judgments.

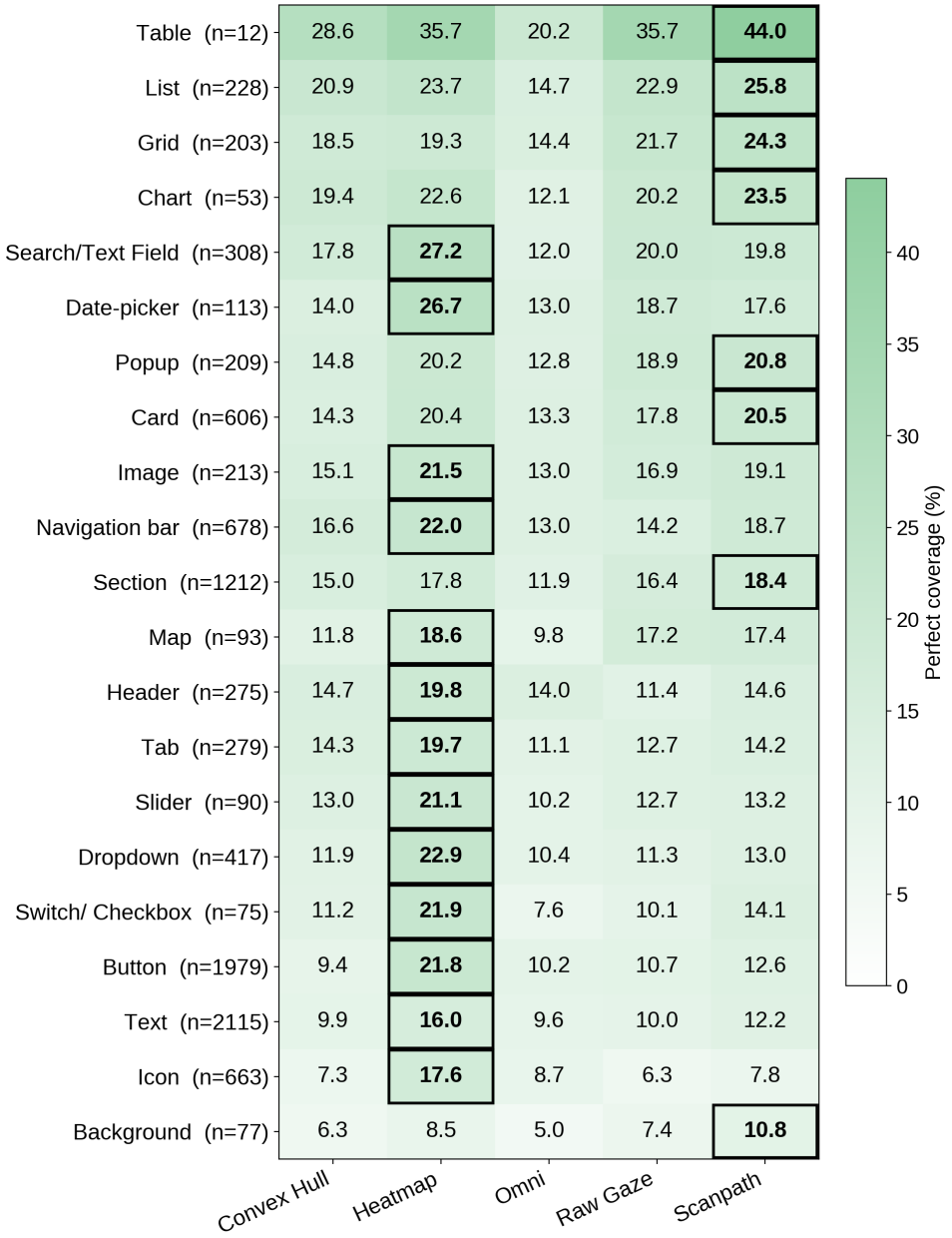



Figure A.3: Perfect Coverage rate across UI types and visualizations.

### A.6 Example of Label Assignment

Original UI images without background shadow for Figure 2a and Figure 2b from Section A.3 are shown below. The following label assignment example corresponds to Figure A.4.

**Ground-truth description**

**User description/annotation:**

"For example the ceramics class begins at 10 a.m, and you can see that it extends almost up to 1 p.m, but not entirely, so you know that it's going to end a little before 1 p.m."

**LLM judge Overall Verdit:**

The user is referring to the ceramics class calendar block whose bottom edge indicates it ends a little before 1 PM.

**Perfect coverage**

**Model Prediction:**

Ceramics for Non-Majors course block (Monday column): The purple/blue calendar block for 'Ceramics for Non-Majors' displayed in the Monday column of the weekly schedule grid, spanning approximately 10 AM to 12:50 PM.,

**LLM judge:**

The prediction identifies that same Ceramics for Non-Majors block spanning roughly 10:00 AM to about 12:50 PM.

**Over coverage**

**Model Prediction:**

Weekly schedule calendar grid with course blocks: The central weekly timetable (Monday–Friday columns) showing colored class blocks around the late-morning time range (about 10AM–1PM).

**LLM judge:**

The prediction selects the broader weekly calendar grid with multiple course blocks across days and times.

**Partial coverage**

**Model Prediction:**

1 PM time label: The text label '1 PM' serving as a reference marker on the vertical time axis of the calendar grid.

**LLM judge:**

The user is referring to the end position of the ceramics class block on the calendar (showing it ends slightly before 1 PM), while the prediction points only to the '1 PM' time-axis label, which is just a reference marker rather than the class block's endpoint."

**Partial+over coverage**

**Model Prediction:**

Weekly course schedule calendar (10 AM - 1 PM time block) and 'Intermediate Studio: Creative Coding' sidebar card: The gaze spans the calendar grid covering the 10 AM to 1 PM rows across Monday through Thursday columns, as well as the 'Intermediate Studio: Creative Coding' course card on the right sidebar panel.

**LLM judge:**

The prediction includes a broad 10 AM–1 PM span across multiple days and also adds an unrelated right-sidebar course card, so it both over-includes other areas and doesn't precisely target the ceramics end time.

**Miss**

**Model Prediction:**

Sidebar card: " 'Intermediate Studio: Creative Coding' course card on the right sidebar panel

**LLM judge:**

Right-sidebar course card is unrelated. There's no meaningful overlap.

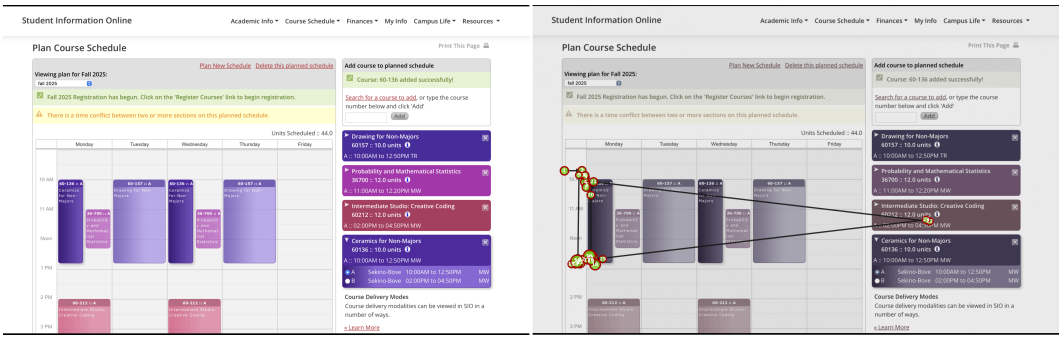



Figure A.4: A scanpath visualization of participant's gaze on a course scheduling platform. Participant's verbal description: "For example the ceramics class begins at 10 a.m, and you can see that it extends almost up to 1 p.m, but not entirely, so you know that it's going to end a little before 1 p.m."

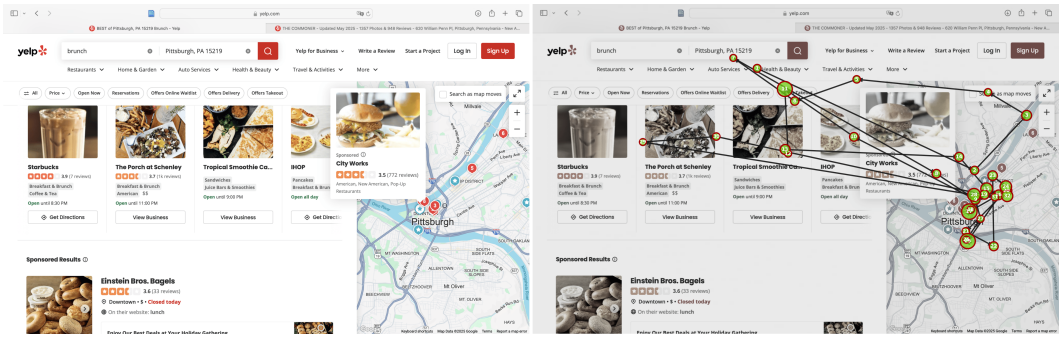



Figure A.5: A scanpath visualization of participant's gaze on Yelp. Participant's verbal description: "And just briefly connect that with what you're seeing under the header, there's a map which the user should be able to navigate. And so, in other words, they can also refine their results visually."

## A.7 Additional Validity Checks

### *A.7.1 Spatial Grounding of Gaze and Language*

To directly examine the spatial correspondence between gaze and concurrent verbal descriptions, we randomly sampled 200 UI elements. One author inspected whether the gaze cluster spatially intersected the element verbalized by the participant. Of the sampled elements, 86.5% showed direct overlap, 11.0% fell on the target's immediate periphery—for example, slightly above or below the element without landing on another element—and 2.5% were completely misaligned. These results indicate that most verbal descriptions are spatially supported by the corresponding gaze data, while also preserving the small peripheral deviations and occasional misalignments present in natural gaze and realistic scenarios.

### *A.7.2 Independent Judge*

GPT-5.2 serves two separate roles in the benchmark: it is one of the seven evaluated prediction models, and separate GPT-5.2 inference calls are used to assign coverage labels. To examine whether this design affects the reported model comparisons and induces self-bias, we randomly sampled 200 UI elements and reran the judge for all 35 model–visualization conditions using Grok-4.3, which is not one of the evaluated prediction models. Grok-4.3 and GPT-5.2 assigned the same coverage label to 5,612 of the 7,000 predictions, corresponding to 80.2% exact agreement.

| Model | Perfect Coverage | | Miss |
|---|---|---|---|
| | GPT-5.2 judge | Grok-4.3 judge | Both judges |
| Gemini-3.1 | 1 | 1 | 3 |
| Sonnet-4.6 | 2 | 2 | 4 |
| **GPT-5.2** | 3 | 3 | 1 |
| Qwen3-235B | 4 | 4 | 5 |
| Qwen3-8B | 5 | 5 | 6 |
| GPT-5-mini | 6 | 7 | 2 |
| Llama-4-Mav | 7 | 6 | 7 |

Table A.3: Model rankings under GPT-5.2 and Grok-4.3 judges, averaged across visualization conditions. Rank 1 indicates the best performance (higher Perfect Coverage and lower Miss).

The two judges produced the same top-five perfect-coverage ordering, differing only in the final two positions, and produced an identical ordering by miss rate, as shown in Table A.3.

The independent judge analysis shows that using GPT-5.2 to judge itself does not have a significant effect on the principal model comparisons, although judge-specific biases cannot be ruled out completely.

### A.8 Additional Baselines

#### *A.8.1 Few Shot Prompting*

The primary benchmark experiments use zero-shot prompting. To examine whether a small number of in-context demonstrations substantially changes performance, we conducted a four-shot pilot on a randomly sampled subset of 200 UI elements. We evaluated Gemini-3.1-Pro-Preview, the strongest model in the primary evaluation, using the two strongest gaze representations, heatmap and scanpath. For heatmaps, perfect coverage changed from 30.5% under zero-shot prompting to 31.0% under four-shot prompting, while the miss rate decreased from 36.5% to 31.0%. For scanpaths, perfect coverage changed from 30.0% to 29.0%, while the miss rate decreased from 30.5% to 28.5%.

The changes in perfect coverage were negligible, while the miss rates decreased modestly, showing GLAZE is a genuinely challenging benchmark. Because this pilot uses only 200 elements and one model, it is not sufficiently powered to characterize few-shot adaptation conclusively. The modest reductions in miss rate motivate larger-scale prompting and fine-tuning experiments to better utilize gaze information.

#### *A.8.2 Human Interpretability of Raw Gaze*

To estimate how much a human can interpret from the raw gaze signal, thus understanding how much LLM can improve to match human performance, we randomly sampled 200 UI elements. One author marked whether they can predict which UI element the user is referring to based on raw gaze. The author rated 69.5% of the examples as confident predictions, 18.5% as possible predictions, and 12.0% as non-predictions. This small analysis indicates that the attended element is often recoverable from the raw gaze representation by a human observer and that substantial room remains for improved model grounding.

### A.9 Gaze Analysis

To characterize how visual attention varies across element categories, we computed seven per-element gaze metrics and examined their intercorrelations. Figure A.6 presents the Spearman rank-correlation matrix for these metrics. Mean fixation duration is largely orthogonal to all quantity-based measures—fixation count, scanpath length, segment time, and word count—which themselves form a tightly correlated cluster.

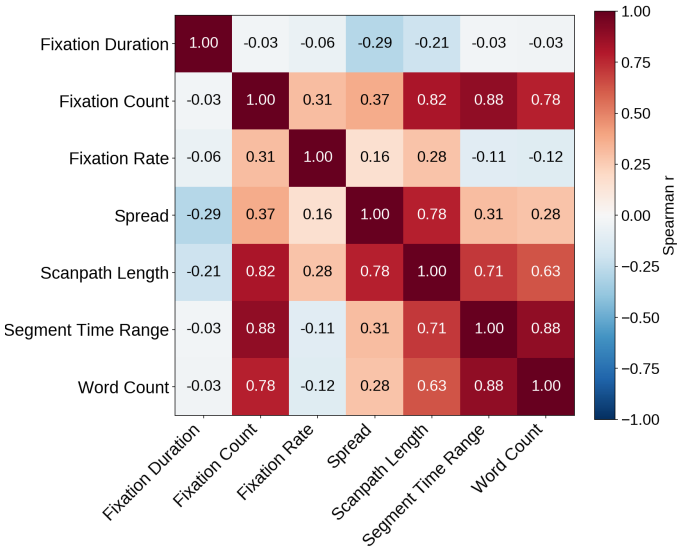



Figure A.6: Spearman intercorrelation matrix across seven per-element metrics.

Figure A.7 presents box plots of mean fixation duration, fixation count, and segment time for each category. The relationship between fixation count and fixation duration is further explored in Figure A.8, which plots the median fixation count against the median mean fixation duration for each category. *Icon* occupies the top-left quadrant, reflecting few but relatively long fixations—consistent with brief yet effortful identification. Container-level categories (e.g., Section, Grid) cluster in the bottom-center region, exhibiting many short fixations characteristic of broad scanning. *Switch/Checkbox* sits at the far right, suggesting prolonged per-fixation processing despite a low fixation count.

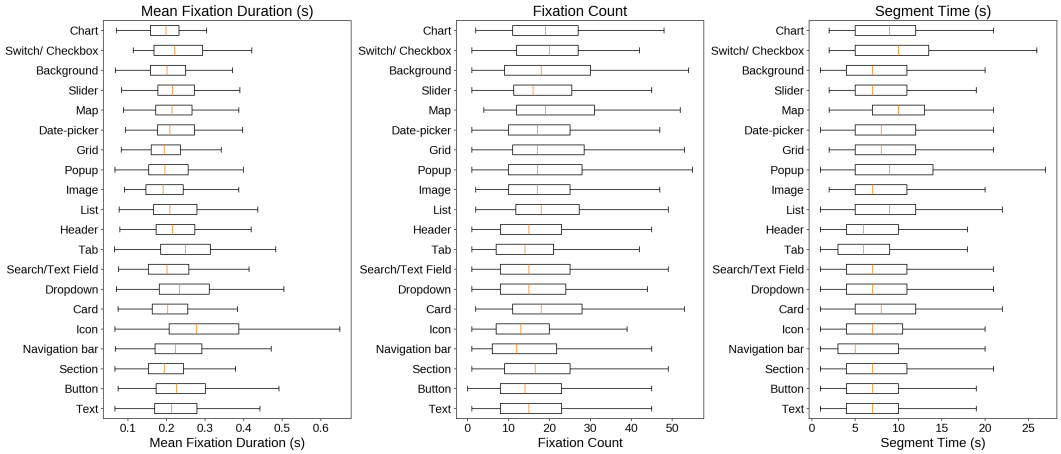



Figure A.7: Box plots (outliers suppressed) of mean fixation duration, fixation count, and segment time per category.

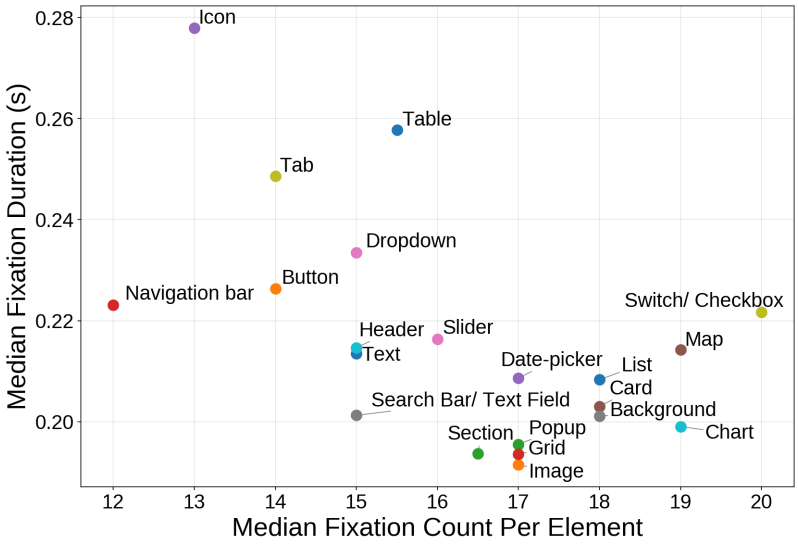



Figure A.8: Median fixation count vs. median mean fixation duration per category.

Figure A.9 shows the fixation spatial spread and bounding-box area for each category. Larger, compositionally complex elements (e.g., Grid, Section) naturally elicit wider spatial dispersion, whereas compact, atomic elements (e.g., Icon, Switch/Checkbox) produce tightly clustered fixation patterns.

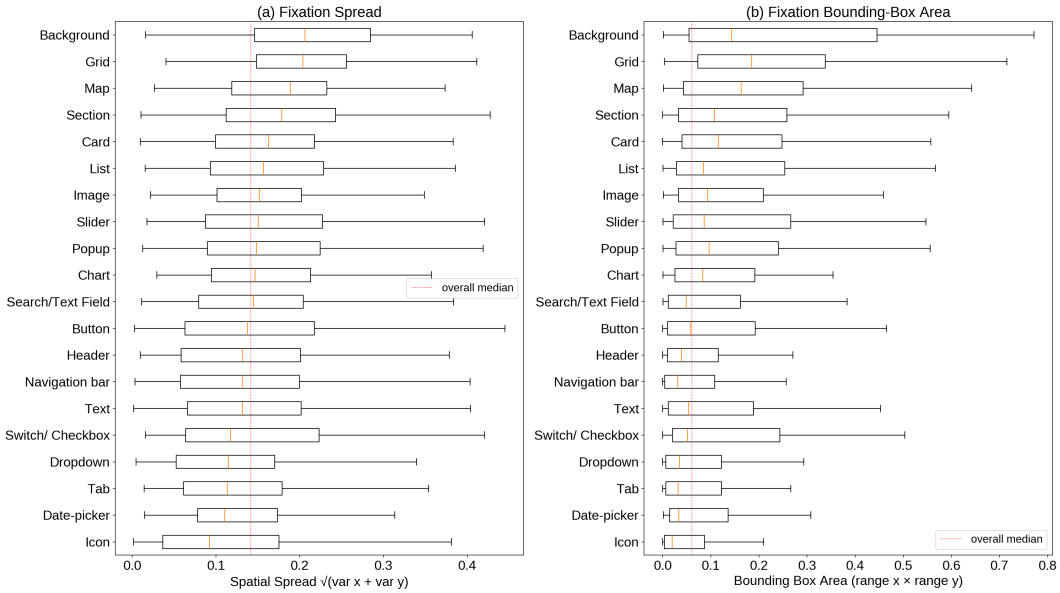



Figure A.9: Fixation diffuseness across UI element categories in screen units. (a) Fixation spread, measured as $\sqrt{\mathrm{Var}(x) + \mathrm{Var}(y)}$; lower values indicate more concentrated gaze. (b) Area of the fixation bounding box, capturing the full spatial range of gaze within a segment.

Finally, Figure A.10 shows per-category fixation heatmaps aggregated over all instances of each category ($y$=0 at the top of the element). Although we do not normalize element size or align elements to a common position, these heatmaps still capture meaningful category-level gaze patterns as well as the typical screen regions where different element types tend to appear. Several of the resulting patterns are consistent with established web-viewing behavior: *List* elements exhibit an F-shaped scanning pattern (Djamasbi et al., 2011), *Text* and other row-structured elements show a strong upper-left concentration of fixations(Djamasbi

et al., 2011), and *Grid* elements display a Z-like traversal pattern suggestive of carousel-style browsing (de Leon-Martinez et al., 2025). At the same time, our benchmark extends the literature by characterizing gaze patterns for UI categories that have received little direct attention in prior eye-tracking work.

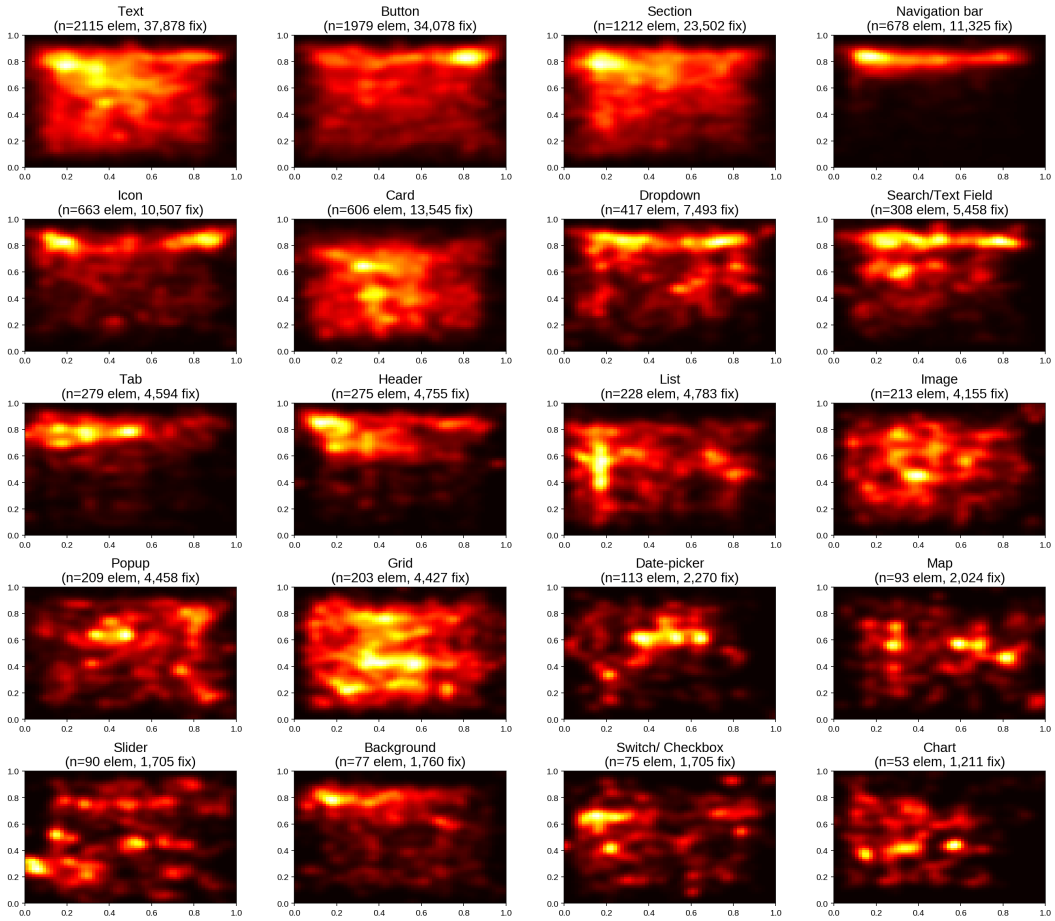



Figure A.10: Per-category fixation heatmaps for all 20 categories ($y$=0 at top). Each panel aggregates all fixations for elements of that category.

### A.10 Prompts

#### *A.10.1 Prompt for LLM Judge*

**Task.** Given a user-described target UI element and an AI-predicted UI element, assign exactly one match label.

**Inputs.** Ground truth (`element_name_hint`, `user_description`), surrounding context, AI prediction (`predicted_element_name`, `element_description`), and screenshot.

**General rules.** `user_description` is the gold standard; if it conflicts with `element_name_hint`, the description wins. Evaluate `predicted_element_name` and `element_description` jointly. Use context and screenshot only to disambiguate the intended on-screen region and compare spatial scope; do not introduce elements not mentioned by the user.

**Labels.**

- **Perfect coverage**: prediction matches the same element(s) at the same granularity.
- **Partial coverage**: prediction is narrower than the target or captures only part of it.
- **Over coverage**: prediction contains the target but extends to a larger or different region.
- **Partial+over coverage**: prediction overlaps with part of the target, but also omits some target content and adds unrelated extra content.
- **Miss**: prediction has no meaningful overlap with the target.

**Procedure.**

1. Identify the target element(s) from `user_description`, using context only to resolve ambiguity.
2. Determine the effective scope of the prediction from the predicted name and description together.
3. Compare the target and prediction in the screenshot in terms of spatial overlap, containment, and granularity, then select the best matching label.

**Special rules.**

- If the user mentions multiple elements, all of them are part of the target.
- A parent container may count as **Perfect coverage** if it stays within the same immediate local area, but as **Over coverage** if it spans distinct page regions.
- Visual properties, primary text labels, and interaction descriptions may be treated as equivalent to the underlying UI element they identify.

### *A.10.2 Prompt for Models Which Predict Human Attention*

**Prompt.**
You are viewing **TWO images** in order:

1. **FIRST IMAGE:** Original UI screenshot (1920×1200 px)
2. **SECOND IMAGE:** [Type of Visualization] overlay showing where the user looked on the same UI
    - Gaze visualization definition (e.g., circle size represents duration for scanpath) and details (e.g., color)

**Task.** Based on the [visualization] in the second image, identify the UI element(s) in the first image that the user was focusing on. Match the natural level of abstraction—e.g., if the gaze covers a navigation bar, report the navigation bar, not its individual buttons; if it covers a single button, report that button.

**Output format.** Respond with **ONLY** the following JSON object, with no additional text:

```
{
"predicted_element_name": "<concise name for the element(s)>",
"element_description": "<one sentence: what the element(s) is and
its location to pinpoint it in the image>",
"reasoning": "<how the gaze pattern led to this identification>"
}
```