Predicting Gaze Location without Camera or Eye-Tracker

Saman Rezapoor,[*] Sajad Shirali-Shahreza,[*†] Gerald Penn[†]
[*]Amirkabir University of Technology
350 Hafez Ave., Tehran, IRAN
saman.rezapour1379.sr@gmail.com, shirali@aut.ac.ir

[†]University of Toronto
4283-40 St. George St., Toronto, CANADA
gpenn@cs.toronto.edu

Abstract

The task of identifying the location that a user looks at, commonly known as gaze estimation, has various HCI and NLP applications. Traditional gaze estimation methods use special hardware such as eye-trackers or ordinary cameras such as webcams to perform this. However, they are not applicable to the majority of web users either because the user does not have them or does not want to use them due to privacy reasons. In this paper, we propose the idea of using multimodal LLMs to analyze the content of the user's screen along with mouse location to estimate the gaze location. It primarily uses the results of studies that extract common reading patterns such as the F-pattern and Z-pattern. Our experimental results on The Eye Of The Typer (EOTT) dataset provide promising results for estimating gaze location.

Keywords: Gaze Estimation, LLM

1. Introduction

Reading online sources such as news, documentation, and long-form articles has become a dominant mode of everyday information consumption. Estimating *where the user is looking* on a webpage, which is usually referred to as gaze estimation or prediction, provides a direct signal of attention. This will enable a range of HCI and NLP applications, including saliency modeling (Buscher et al., 2009), adaptive interfaces, and improved placement (or suppression) of distracting elements(e.g., advertisements) (Owens et al., 2011).

Gaze prediction data also offer a valuable supervision signal for language understanding tasks, because reading behavior reflects how users allocate attention across words, headings, and images. Despite its value, gaze estimation remains difficult to deploy at scale: high-quality eye trackers are expensive, sensitive to setup conditions, and rarely available in naturalistic browsing settings.

To reduce cost and deployment friction, previous work has studied *proxies and constraints* for gaze estimation while the user is browsing. Reading often produces structured scan-paths shaped by page layout (commonly described by patterns such as *F-shaped* or *Z-shaped* scanning). These regularities can be leveraged to constrain gaze inference (Soegaard, 2021; Lorigo et al., 2008). Other studies have analyzed the relation between gaze location and interaction logs. They showed that mouse movements and clicks correlate with attention during navigation and reading (Huang et al., 2012; Navalpakkam et al., 2013). They used classical machine learning models to predict gaze from such signals.

Webcam-based systems such as WebGazer[1] further illustrate that gaze can be predicted without dedicated hardware, though accuracy can vary with calibration and user/environmental factors (Papoutsaki et al., 2018).

Motivated by these precedents, we tried to determine whether *multimodal large language models (LLMs)* can predict gaze in a naturalistic browsing setting *without* any special hardware such as eye trackers or even webcams. Our idea focuses on previous findings in regard to reading behaviour and temporal continuity. We report our preliminary results in this paper. Concretely, our contributions are:

1. Prompt-driven gaze prediction: encoding reading scan-pattern priors and temporal smoothing constraints into a system prompt and enforcing *structured JSON outputs* (coordinates, attention-pattern label, reasoning-mode label, and confidence).
2. Two evaluation conditions: *full-video* prediction from screenshot sequences, which does not need any actual gaze location data, and a *per-frame setting* that conditions each prediction on the actual gaze location in the previous frame. [2]

[1] webgazer.cs.brown.edu/

[2] The second condition aims to remove the effect of previous prediction errors (accumulated error) in predicting the gaze in the next frame, i.e., isolating the effect of temporal anchoring.

Proceedings of The Second International Workshop onEye-Tracking Resources and Evaluation for Human-Aligned NLP, pages 58–63
May 12, 2026.

3. Quantitative metrics and analysis: We report *pixel-level error* and *region-level accuracy*, and analyze how model-reported confidence relates to error; this includes failures in low-information cases such as empty and calibration pages.

2. Related Work

Our work lies at the intersection of (i) mouse-based proxies for gaze and attention, and (ii) learning-based models of gaze and reading behavior. We briefly review both areas and position our contribution in relation to them.

2.1. Mouse Proxy for Gaze/Attention

Human-computer interaction has long grappled with the challenge of whether the cursor trace, including mouse movements, hovers, and clicks, can be used to estimate where the user is looking and what they are attending to.

Huang et al.'s (2012) work on web search concerns the context dependency of gaze and cursor alignment and demonstrates how gaze prediction benefits from the inclusion of behavioral features rather than just the cursor position. Perhaps most interestingly, their results suggest that the lag between the two is consistent, with the cursor lagging behind the gaze by close to $700ms$ on average, which in turn supports the intuitive idea that users tend to look ahead of their mouse movements (Huang et al., 2012; Milisavljevic et al., 2021).

Related results have also been presented in search and browsing settings where eye and mouse movements are modeled explicitly over time and page structure. Navalpakkam et al. (2013) demonstrate that, in non-linear page structures, mouse movements can be used for predicting attention patterns, even though they also exhibit systematic discrepancies depending on the task and state of interaction. Guo and Agichtein (2010) also present results on gaze prediction based on mouse movements in web search settings. They emphasize the importance of inferring the dynamic coordination between gaze and mouse movements.

More recently, Popescu and Nisioi (2025) propose the Mouse Tracking for Reading (MoTR) method for predicting the reading time. They blurred the text except for under the mouse cursor, which allows them to estimate reading times based on mouse movements. The authors show that the reading times obtained with MoTR capture standard psycholinguistic effects and are predictable using lexical features and transformer models. Although MoTR provides evidence for the viability of mouse tracking for reading analysis, it does not directly validate mouse traces against simultaneous eye tracking, nor does it attempt gaze estimation. Therefore, it is not trying to predict gaze from mouse tracking data, which is our main idea in this paper.

2.2. Learning-based modeling of gaze and reading behavior

A second line of work involves the use of machine-learning methods for modeling eye-tracking outcomes from linguistic and interactive features. Alves (2025) benchmarks LLM-based methods for predicting eye-tracking reading-time measures (first-fixation duration, gaze duration, total fixation time). Their results show a high variance in predicting such values. That work focused on temporal effort signals (durations), however, rather than predicting the gaze (spatial gaze coordinates) on rendered pages.

Another proposed idea for performing scalable gaze estimation without the use of eye-tracking devices is to use webcams for gaze prediction. Papoutsaki et al. (2016) demonstrate the viability of the method using the WebGazer tool, which utilizes self-calibration based on user interactions for the estimation of the gaze location using commodity-grade webcams. They also released The Eye Of The Typer Dataset (EOTT)[3], which provides a collection of synchronized screen recordings, mouse movements, and eye-gaze locations for different tasks (Papoutsaki et al., 2018).

Recently, Ahmadzadeh (2024) used classical supervised methods to predict gaze location or areas-of-interest (AoI) using interaction traces (mouse and keyboard events) only. The results show that **predicting AoI** is much more accurate than predicting exact coordinates. The present paper is one example of such work (Ahmadzadeh, 2024).

2.3. Positioning of the present work

In comparison with related gaze prediction studies, our approach does not rely on a uniform mouse-gaze mapping but rather uses mouse traces as an additional cue. It emphasizes layout-driven reading priors and temporal continuity. Compared with ML/NLP methods that predict reading-time-related quantities (Michaelov and Levy, 2026), we predict spatial gaze location from screenshots. We use two metrics to measure our accuracy: pixel-wise error and region-wise accuracy. As to methodology, we also examine how the accumulated errors of multimodal LLMs during gaze prediction reduce accuracy over the duration of the task.

[3] https://webgazer.cs.brown.edu/data/

3. Our Proposed Method

We propose a prompt-driven gaze prediction framework that targets an article-reading scenario. The framework (i) specifies attention-pattern priors and temporal smoothing constraints through an LLM prompt, and (ii) evaluates the resulting model behavior on a sequence of screenshots sampled from long interaction traces. Because the prompt is tailored specifically article reading, we filter the available frames to article-like pages and then obtain model predictions for each frame in chronological order.

3.1. Dataset

We base our experiments on **The Eye of the Typer (EOTT)**[4] dataset, released with Papoutsaki et al. (2018). This dataset provides synchronized screen recordings, eye-tracing logs (i.e., actual gaze location), and interaction traces from 51 participants during multiple web-based tasks.

From the full dataset, we randomly selected 3 users and used the per-user metadata in `Participant_Characteristics.csv` to obtain session durations and native screen resolutions. The gaze coordinates are represented in pixel space in the dataset logs.

3.2. Extraction and synchronization pipeline

To create the input prompts for the multimodal LLM, we created a pre-processing pipeline based on the EOTT "dataset extractor"[5] utility. We adapted it to extract (i) screenshots, (ii) gaze samples aligned to each screenshot, and (iii) mouse events rendered directly onto the screenshot images.

3.2.1. Screenshot sampling

We extracted video frames as screenshots at a fixed 5-second interval. This rate was selected to avoid excessive oversampling while still preserving coarse reading progression. We selected 50 screenshots in total from each participant. The main emphasis was on segments where participants were reading article-like pages, plus a small number of sparse/empty pages (e.g., calibration or low-content transitions) to test robustness under low visual density.

3.2.2. Gaze log alignment

For each screenshot captured at timestamp t, we selected gaze samples from `[Dataset_Name].txt` whose timestamps fall within a temporal window of $\pm 500\text{ms}$ around t. The gaze stream in our extracted logs is sampled at 60 Hz, as stored in the dataset traces used by our pipeline. For each screenshot, we retained up to 5 gaze points closest to t (a fixed window size), yielding a compact set of candidate gaze locations per frame.

3.2.3. Mouse event extraction and rendering.

Mouse events were read from `[Dataset_Name].json`. We rendered interaction traces onto screenshots using OpenCV[6]: (i) a polyline for mouse movement and (ii) a filled red circle at click locations. This augmentation was motivated by earlier findings that mouse activity can correlate with visual attention in some interaction modes, while remaining an imperfect proxy overall (Huang et al., 2012; Navalpakkam et al., 2013).

3.3. Reference gaze per screenshot via geometric median

Because raw gaze samples may include transient noise, drift, or occasional rapid eye saccades, we aggregate the windowed gaze points into a single *reference gaze coordinate* per screenshot using a geometric median (i.e., Fermat–Weber point). Concretely, for gaze samples $\{p_i\}_{i=1}^{n}$ in a screenshot window, we estimate:

$$\hat{p} = \arg\min_{p} \sum_{i=1}^{n} \|p - p_i\|_2.$$

We compute $\hat{p}$ using Weiszfeld's algorithm, iterating until it converges (Beck and Sabach, 2015).

We selected the geometric median instead of an arithmetic mean because it is more robust to outliers: when a participant's gaze briefly "jumps" (e.g., due to a saccade or tracking noise), the median reduces the influence of those samples relative to the mean. In our implementation, this robustness serves as an implicit outlier-mitigation step rather than applying a separate rejection rule. [7]

3.4. Prompted gaze prediction with multimodal LLMs

We evaluate two different multimodal LLMs as **prompted gaze predictors** conditioned on screenshot content and interaction overlays. Our prompts

[4] https://webgazer.cs.brown.edu/data/

[5] https://github.com/brownhci/WebGazer/

[6] https://opencv.org/

[7] Indeed, another view of gaze, as pointed out by an anonymous reviewer, is that attention is established through time and is mediated through an explicit temporal window over which probability distributions of gaze and mouse location could both be calculated. Our smoothing constraint is a crude approximation of this more sophisticated model.

encode established web-reading heuristics, particularly layout-driven scanning patterns such as the F-pattern and Z-pattern, to constrain the model towards plausible reading behavior over article-style pages (Huang et al., 2012; Navalpakkam et al., 2013).

3.4.1. Models

We tested OpenAI's GPT-5.2 Instant and Google's Gemini 3 Flash (preview) as representative of current state-of-the-art multimodal LLMs capable of image understanding. Inputs were downsampled and compressed prior to submission to comply with API limitations in processing screenshots. The gaze predictions are mapped back to screenshot coordinates later on. The output predictions were also clipped to valid screen bounds.

3.4.2. Structured output constraint

Each inference produces a strictly valid JSON:

- (x, y) pixel coordinates
- `attention_pattern` $\in$ {`F-pattern`, `Z-pattern`, `Center-focus`, `Cursor-aligned`}
- `reasoning_mode` $\in$ {`Reading`, `Scanning`, `Targeting`, `Idle`}
- `confidence` $\in [0, 1]$

A schema-based constraint is used to reduce free-form text and enforce machine-readable outputs for downstream evaluation.

4. Experimental Results

4.1. Experimental Scenarios

We define two evaluation conditions or scenarios:

1. **Full-video prediction.** The model receives only the screenshot sequence (with mouse overlays) and produces a gaze estimate and metadata for each frame. No ground-truth gaze is provided during the run.

2. **Per-frame prediction.** After the model predicts the gaze for frame t, we immediately provide the actual reference gaze coordinate $\hat{p}_t$(geometric median) as ground truth. The next frame's prompt explicitly instructs the model to use this provided coordinate rather than its previous prediction. This setting provides an online correction signal, analogous to teacher forcing for sequence prediction, to prevent error accumulation as the model predict frames. This enables us to test the model's power in predicting the next gaze location. As a collateral benefit, it enforces **temporal continuity** in that it ensures smooth gaze trajectories across 5-second steps.

4.2. Numerical Results

We tested two multimodal LLMs under identical conditions: *Google's Gemini 3 Flash (preview)* and *Open-AI's GPT-5.2 Instant*. Tests were conducted on data from three randomly selected participants from the EOTT dataset (Papoutsaki et al., 2018) (P1, P2, and P7) and under two scenarios: full-video and and per-frame.

To calculate region-level estimation accuracy, we divided the screen into 300x300 pixel areas and checked whether the predicted gaze location is in the same region as the actual gaze location. This evaluation regimen is commonly used in previous work (e.g., (Ahmadzadeh, 2024; Papoutsaki et al., 2018)), because some applications only need the rough location of the gaze.

4.2.1. Model Comparison

GPT-5.2 Instant yields more stable predictions than Gemini 3 Flash. The results of these two models for one of the selected participants (P1) are shown in Table 1. The results are very close, both in terms of region accuracy and gaze coordination error, although GPT-5.2 Instant is slightly better. A key qualitative difference is that GPT-5.2 Instant tends to lower its confidence when page content provides insufficient evidence (e.g., sparse or ambiguous layouts), whereas Gemini 3 Flash often maintains moderate confidence in such cases.

Therefore, in the analysis below, we only show the results for GPT-5.2 Instant.

4.2.2. Overall results

Table 2 reports detailed results for the tested users, as well as macro-averaged results.

Our first observation is the relatively low accuracy in estimating the gaze region (around 17% for Full-Video and around 24% in the Per-Frame scenario). However, when the region is not accurately estimated, in more than half of the cases, the estimated region was one of the adjacent regions (up, down, left, or right). In other words, the estimated region is not very far from the actual location of the gaze in the majority of the cases.

As we expect, the estimation errors accumulate in the Full-Video scenario and therefore the performance is worse: around 50 pixels higher for estimated location error and around 10% lower region accuracy. This trend is consistent for all participants.

Model	Condition	Region Accuracy	Average Error (px)	Median Error (px) (All)	(CL>0.8)	(CL>0.9)
Gemini 3 Flash	Full-Video	15.7%	357	296	293	307
	Per-Frame	27.5%	290	229	228	224
GPT-5.2 Instant	Full-Video	17.7%	356	322	321	322
	Per-Frame	27.4%	286	210	196	210

Table 1: Comparison of different multimodal LLMs.

User	Condition	Average Error (px)	Median Error (px) (All)	(CL>0.8)	(CL>0.9)	Region Accuracy (%) (All)	(CL>0.8)	(CL>0.9)
P1	Full-Video	356	322	321	322	17.7	16.2	16.2
	Per-Frame	286	210	196	210	27.4	25.6	25.6
P2	Full-Video	357	341	341	341	17.6	17.6	17.6
	Per-Frame	295	253	253	262	27.5	30.4	30.4
P7	Full-Video	428	411	411	446	15.7	10.7	10.7
	Per-Frame	410	385	403	414	17.6	14.3	14.3
Average	Full-Video	380	358	358	370	17.0	14.8	13.5
	Per-Frame	330	283	284	295	24.2	23.4	23.4

Table 2: Gaze prediction results of our method for different users.

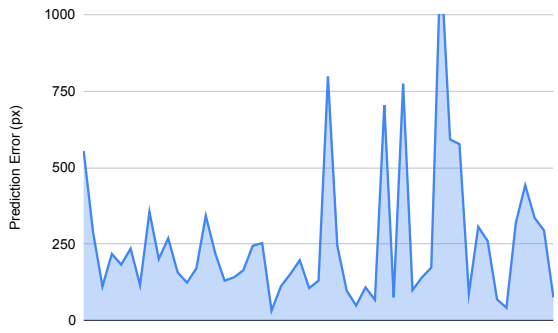

Figure 1: Per-frame error spikes of one user.

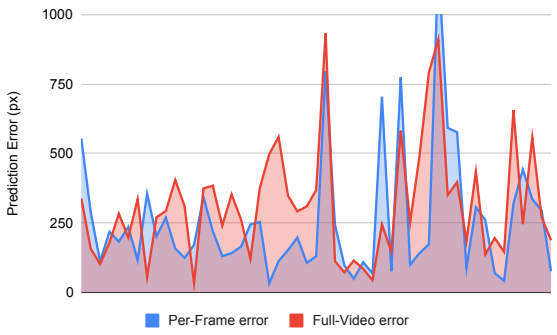

Figure 2: Comparison of per-frame and full-video error rates.

We also see that while the performance is similar for the first two users, it is significantly worse for the third user, especially in the per-frame scenario. This can be due to individual differences between human users.

4.2.3. Temporal error dynamics

Figure 1 plots the per-frame error over time for one of the users (P1). Error spikes align with context discontinuities, such as navigation events or transitions to empty/calibration-like pages. In such frames, the model lacks strong layout cues (text blocks, headings, images) needed to infer reading patterns, leading to unreliable gaze estimates. Moreover, empty-page segments can affect subsequent predictions by disrupting the inferred scanpath trajectory, especially in the full-frame scenario, in which errors accumulate.

4.2.4. Effect of per-frame prediction

Figure 2 contrasts the full-video and per-frame scenarios for one user (P1), illustrating that providing the previous ground-truth gaze substantially stabilizes trajectories after context shifts by preventing error accumulation. This suggests that temporal anchoring mitigates drift and reduces cascading errors when visual evidence is weak.

5. Conclusion

In this paper, we have proposed the idea of using multimodal LLMs to predict gaze using only a screenshot and the movement of the mouse pointer. Our preliminary results show that, while our results are not very high, they are promising. For example, our approach can correctly guess either the correct region or a region adjacent to it (among 12 300px x 300px regions on the page) in over half of the

cases we tested. We think with the advances in small multimodal LLMs (such as small Gemma 3 variants that can easily be run on user devices), our idea, with refinement, can serve as the basis for on-device gaze estimation without further need for any special hardware.

Limitations

The main motivation of our work was to remove the need for special hardware (such as eye-tracker or cameras) in gaze estimation. Such hardware typically requires continuously watching the user with a camera, which is a significant privacy concern. Our approach only needs to analyze the content of the screen. It is therefore less intrusive. On the other hand, it requires submitting a screenshot to an LLM.

A limitation of our current results is that it is based on a public dataset that was collected in a lab setting. We should follow up with more evaluations that run in a user's setting (e.g., in their home or office). We should also evaluate the accuracy of gaze prediction with techniques that are usually used during calibration.

Acknowledgements

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Seyed Fatemeh Ahmadzadeh. 2024. User's gaze estimation based on the movement of the mouse and screen information. Bachelor's thesis, Amirkabir University of Technology, Tehran, IRAN, June.

Diego Alves. 2025. Benchmarking language model surprisal for eye-tracking predictions in Brazilian Portuguese. In *Proceedings of the First International Workshop on Gaze Data and Natural Language Processing*, pages 7–17, Varna, Bulgaria. INCOMA Ltd., Shoumen, BULGARIA.

Amir Beck and Shoham Sabach. 2015. Weiszfeld's method: Old and new results. *J. Optimization Theory and Applications*, 164(1):1–40.

Georg Buscher, Edward Cutrell, and Meredith Ringel Morris. 2009. What do you see when you're surfing? using eye tracking to predict salient regions of web pages. In *Proceedings of CHI*, pages 21–30.

Qi Guo and Eugene Agichtein. 2010. Towards predicting web searcher gaze position from mouse movements. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, page 3601–3606. ACM.

Jeff Huang, Ryen White, and Georg Buscher. 2012. User see, user point: gaze and cursor alignment in web search. In *Proceedings of CHI*, pages 1341–1350.

Lori Lorigo, Maya Haridasan, Hrönn Brynjarsdóttir, Ling Xia, Thorsten Joachims, Laura Granka, Fabio Pellacini, and Bing Pan. 2008. Eye tracking and online search: Lessons learned and challenges ahead. *JASIST*, 59:1041–1052.

James A. Michaelov and Roger P. Levy. 2026. N-gram-like language models predict reading time best. Technical Report 2603.09872, arXiv.

Alexandre Milisavljevic, Fabrice Abate, Thomas Le Bras, Bernard Gosselin, Matei Mancas, and Karine Doré-Mazars. 2021. Similarities and differences between eye and mouse dynamics during web pages exploration. *Front. Psychol.*, 12:554595.

Vidhya Navalpakkam, LaDawn Jentzsch, Rory Sayres, Sujith Ravi, Amr Ahmed, and Alex Smola. 2013. Measurement and modeling of eye-mouse behavior in the presence of nonlinear page layouts. In *Proceedings of the 22nd Int. Conference on World Wide Web*, pages 953–964.

Justin W. Owens, Barbara S. Chaparro, and Evan M. Palmer. 2011. Text advertising blindness: The new banner blindness? *J. Usability Studies*, 6(3):172–197.

Alexandra Papoutsaki, Aaron Gokaslan, James Tompkin, Yuze He, and Jeff Huang. 2018. The eye of the typer: a benchmark and analysis of gaze behavior during typing. In *Proceedings of ACM ETRA*.

Alexandra Papoutsaki, Patsorn Sangkloy, James Laskey, Nediyana Daskalova, Jeff Huang, and James Hays. 2016. Webgazer: scalable webcam eye tracking using user interactions. In *Proceedings of IJCAI*, pages 3839–3845. AAAI Press.

Cristina Maria Popescu and Sergiu Nisioi. 2025. Exploring mouse tracking for reading on Romanian data. In *Proceedings of the First International Workshop on Gaze Data and Natural Language Processing*, pages 44–51, Varna, Bulgaria. INCOMA Ltd., Shoumen, BULGARIA.

Mads Soegaard. 2021. Visual hierarchy: Organizing content to follow natural eye movement patterns. IxDF - Interaction Design Foundation (accessed 29th March, 2026).