Your Mouse and Eyes Secretly Leak Your Preference: LLM Alignment using Implicit Feedback from Users

Anonymous ACL submission

Abstract

To align a Large Language Model (LLM), most existing methods collect explicit human feedback and train a reward model to predict the human preference based on the response text. These existing methods have two key limitations. First, the users rarely provide explicit feedback for LLM responses, which makes the high-quality preference annotation expensive to collect. Second, the methods do not leverage implicit human feedback, which has proven vital to the economic moats of Internet giants. To quantify the value of implicit feedback, we build a new dataset called IFLLM, which collects 1336 multi-turn questions from the 59 Mechanical Turk workers, their mouse trajectories, and eye gazing points to the LLMs' responses from their webcams. IFLLM shows that the users have very diverse types of gazing behavior and mouse trajectories. Our reward model based on the implicit user feedback boosts the accuracy of the text-based reward model from 55% to 64% and nearly triples the relative response quality improvements after applying the DPO to eight LLMs, demonstrating the value of implicit feedback in the wild. Our data collection website, dataset, and codes can be found at `https://anonymous.4open.science/r/llm_implicit_feedback/`.

1 Introduction

Large-scale intelligent systems deployed in industry are designed to satisfy user needs and align with user expectations. In early stages of system development, researchers and practitioners typically rely on assumptions about these expectations, informed by prior experience and limited user interviews. These assumptions are subsequently operationalized through the design of annotation guidelines and the collection of labeled data used to optimize system performance. However, this paradigm does not scale effectively. Large-scale human annotation is both costly and time-intensive, and the resulting data often fails to accurately reflect real-world user interactions.

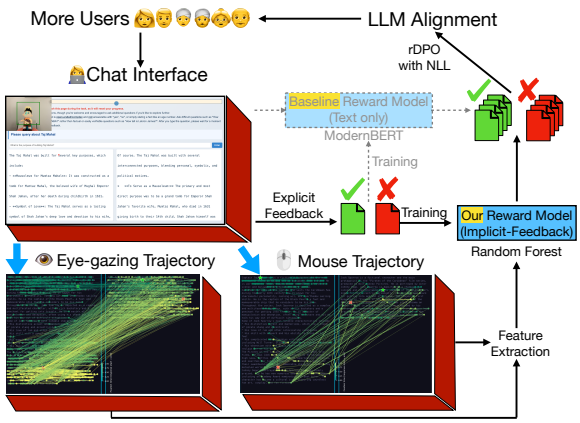


Figure 1: IFLLM records the trajectories of eye gazing and mouse from a question answering session between a user and two LLMs. Then, we train our random forest reward model on the features extracted from the trajectories and preference labels from the user. Finally, we show that applying DPO to preferences predicted by our reward model improves LLM outputs more than a standard text-based reward model. This improvement could attract more users, enrich implicit user preferences, and promote a positive feedback loop.

To address this limitation, some user-facing LLM providers, such as OpenAI, incorporate explicit user feedback on generated responses (Han et al., 2025). This strategy is particularly important given that only 1–3% of users provide feedback such as thumbs-up or thumbs-down (Wang et al., 2025b). Moreover, prior work suggests that frequent solicitation of explicit feedback can negatively impact user satisfaction (Zhao et al., 2018).

This is why prior successful intelligent systems, such as search engines and recommender systems, have extensively used implicit feedback signals for improving their system. For instance, click data is an important signal (if not the most important signal) in training ranking models in search engines (Joachims, 2002) and recommender systems (Oard

and Kim, 1998). Despite tremendous success in using implicit feedback in these technologies, implicit feedback has been relatively underexplored for improving user-facing LLM technologies. A main reason is that common implicit feedback signals used in prior systems, such as clickthrough data, barely exist in many of these systems (Allan et al., 2024). In other words, users infrequently engage with links provided by LLMs, when such links are available, and even when these interactions occur, it remains unclear how to reliably interpret them as training signals. *Therefore, this paper investigates implicit feedback in the context of user-facing LLM systems and examines its potential for improving model alignment.* In this work, we focus on two forms of implicit feedback: (1) *mouse movement*, which is readily available at scale in real-world deployments, and (2) *eye tracking*, which, although not yet widely accessible, is representative of a broader class of multimodal user signals. We anticipate a future in which intelligent assistants can leverage such inputs, including gaze patterns, facial expressions, and hand gestures, to better model user intent and improve system alignment. To improve the practical feasibility of eye tracking signals, we rely on webcams, as opposed to special-purpose eye trackers, which are often only available in controlled lab environments. We have performed extensive efforts in developing a webcam-based crowdsourcing website that can be calibrated per user to work effectively with different cameras, internet browsers, and screen sizes and resolutions. The developed crowdsourcing website is released publicly for future use.

Building upon our developed website, we first collect a new dataset, IFLLM (**I**mplicit **F**eedback for **L**arge **L**anguage **M**odels), which contains 1336 multi-turn question-answering interactions collected from 59 unique Amazon Mechanical Turk workers across hundreds of topics from Wikipedia. During each interaction, users choose the topics they want to learn about, ask at least three questions, either score a single response (pointwise setting) or compare a pair of responses from different LLMs (pairwise setting), and answer some post-task questions. During the study, users' mouse movements and gaze trajectories are continuously recorded, with their consent.

Using IFLLM, we first analyze how users read and evaluate LLM responses. We show that user behavior is highly diverse and strongly influenced by response length. For example, mouse trajectories become increasingly correlated with user gazing trajectories for long responses because users must scroll through the generated response to read further. Our experiments show that the features extracted from the mouse trajectories are essential to our random forest preference classifier while the gazing signal is helpful when the responses are short. Through SFT and DPO, we train the LLMs to produce the responses that are more likely to be pointed by users' mouse and thus receiving higher scores from an LLM judge and human annotators.

MAIN CONTRIBUTIONS

- We build IFLLM, the first dataset that contains mouse and eye-gazing trajectories as well as explicit user preference for LLM responses in a realistic, multi-turn conversational setting. We also release our website source code for collecting IFLLM under Apache 2 license to facilitates future implicit feedback data collection.
- This paper provides the first systematic in-depth analysis of diverse reading behaviors and compares the effectiveness of mouse and gaze signals for training reward models.
- We show that implicit feedback, especially mouse movement for longer responses, substantially boosts preference prediction accuracy and raises the DPO improvements from 0.12 to 0.35.

We posit that our findings, together with the released website and dataset, lay the groundwork for next-generation alignment methods that not only improve average LLM performance, but also enable scalable personalized alignment, which is known to significantly enhance user satisfaction (Salemi et al., 2024).

2 Related Work

Implicit feedback has been shown to be a valuable signal. Researchers have improved LLMs through human edit (Gao et al., 2024), human responses in a multi-turn conversation (Shi et al.; Wang et al., 2025b), and click and copy behavior (Wang et al., 2026). However, they do not study the effectiveness of the eye-gazing and mouse signal.

Eye gazing data could help (large) language models in many different ways. It could be used to estimate the weights of each token in supervised fine-tuning (SFT) (Zhang et al., 2025), align the attention of the transformer with human attention (Zhang et al., 2024; Yan et al., 2024), rearrange the order of aggregating contextualized embeddings (Deng et al., 2024), guide LLMs to

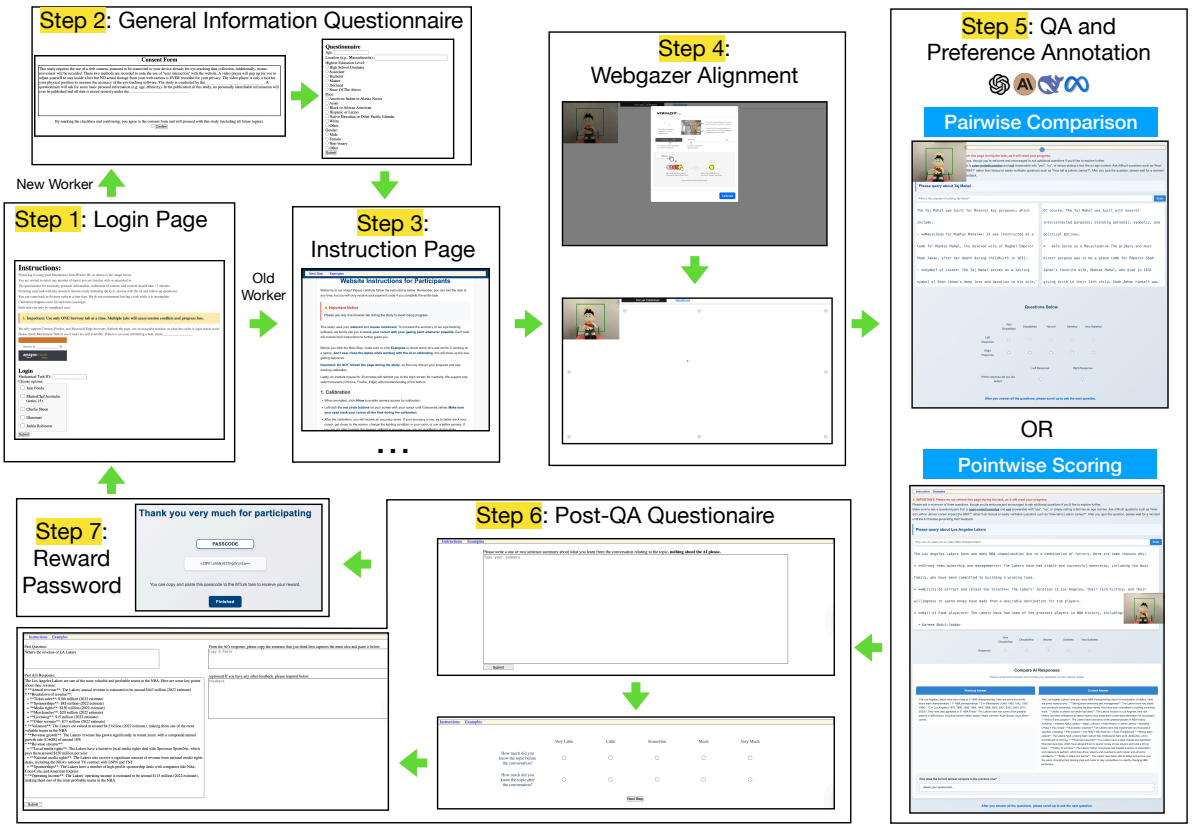


Figure 2: Diagram of webpage navigation for a worker. 1 cycle of the webpages correlates to 1 task, equivocally 1 topic was conversed and annotated. Steps 1-4 prepare the user for a task. We record the eye-gazing and mouse movement data in Step 5. After the user complete the questionaires in Steps 6, they can use the password in Step 7 to claim their reward in MTurk.

generate text with different readability (Säuberli et al., 2026), study how humans collaborate with LLMs (Tang et al., 2024b,a), and improve the reward model and LLM alignment (Lopez-Cardona et al., 2025b; Papadopoulos, 2025). However, no study collects and compares the implicit user feedback signals on LLMs' responses in the wild.

Eye-gazing data has many applications in natural language processing (Mathias et al., 2020) and machine learning. For example, eye-gazing data can be used to predict the linguistic acceptability (Bondar et al., 2025b,a), predict the image preference of humans (Papadopoulos et al., 2026), and analyze the interaction of humans with coding agents (Yang et al., 2025b; Wang et al., 2025a). However, these works do not focus on humans' implicit feedback to the LLMs' answers. One notable exception is the OASST-ETC dataset (Lopez-Cardona et al., 2025a), which collects clean eye-gazing data in a controlled laboratory setting. Nevertheless, their reliance on special eye-tracking equipment and the neglect of valuable mouse movement data make them unsuitable for investigating whether LLMs could benefit from the usage of the general public.

3 The Data Collection Website

We develop a website for users to converse with LLMs. Users are recruited from Amazon Mechanical Turk (MTurk) under an approved Institutional Review Board (IRB) protocol. Our website allows a MTurk worker to do our tasks using multiple windows/tabs in Google Chrome, Firefox, or Microsoft. Steps 1-7 of Figure 2 represent one run through of a task with one selected topic.

3.1 Login and Personal Questionnaire

Step 1 is the Login Page where the worker selects the topic(s) they want to know more from a pool of 30 or 60 topics and we shuffle the topic order to avoid positional bias. Each topic is a Wikipedia page title chosen from the bottom of top 1000 popular search results from the Wikimedia API between 1/2023 and 5/2023. Our strategy aims to find the topics that the users have heard of but are not very familiar with.

An input field requires the MTurk Worker ID.

A new user will get redirected to Step 2: General Information Questionnaire, which asks the user to consent our data collection and provide some demographic information, or Step 3: Instruction Page for returning users. Throughout this experiment, we ensure instructions are accessible and clear. For each session, the user must calibrate at Step 4: Webgazer Alignment. Webgazer.js (Papoutsaki, 2015) tracks your eye movement from the webcam and predicts your gazing points using a regression model. We use the calibration tool of Webgazer to train the eye-gazing model. The Webgazer displays the user's camera and instructs the user to position his/her head inside a green box for a better tracking accuracy.

3.2 QA and Preference Annotation

Step 5 carries out the LLM conversation through the QA and Preference Annotation pages. Each topic is assigned as a pointwise scoring or pairwise comparison task. The user is instructed to ask non-factual questions to know more about the selected topic. Pairwise comparison uses two textboxes side-by-side. Pointwise scoring only uses one textbox. Each LLM response box is a random choice from DeepSeek V3, GPT-4o Mini, Claude Sonnet 4.5 (originally 3.5 but deprecated), or Llama 3.3 70B (Grattafiori et al., 2024) with no duplicates in the pairwise setting. The LLMs were chosen for their popularity, diversity, significance, and/or being open-sourced. We also want to check if the structure of the LLM response affects the users' gazing pattern, so we randomly instruct the LLM to reply using bullet points.

The user is asked to query at least three times and spend at least 90 seconds in this step. LLMs remain the same for each topic across all queries and can access the prior queries and answers, which allows the user to ask follow-up questions or conduct multi-turn interactions. If an LLM response is too long, it overflows the textbox with scrolling in the textbox enabled. At approximately every 0.1 seconds, we record the character index and coordinates of gaze and mouse positions. Under the LLM response(s), there are question(s) for the 5-point Likert quality scale and preference annotations of either they prefer the previous LLM response compared to the current in the pointwise setting or which response is preferred in the pairwise setting. The worker must finish all the annotations before asking the next question. The size of the textbox, font, and line spacing are large for user readability and better eye-gazing accuracy.

3.3 Post-Test Questionnaires

Step 6 is a Post-QA Questionnaire over three pages. The first page asks the user for a brief summary of the conversation for quality control. The second page questions the user with Likert scales (1-5) on the user's knowledge of the task before and after to quantify the quality of the LLM conversation. The last page provides the user opportunity to give feedback while asking to copy a sentence to test if a user gazes at sentences they deem significant. Step 7 gives a password to submit in Mechanical Turk as the final verification step that the user completed a task. If the worker choose multiple topics in Step 1, the user would directly go to Step 5 for the next topic after completing Step 7, which avoids wasting time on constantly gazing calibration.

3.4 Quality Control

The first step for quality control is a minimum accuracy threshold of 70%, a tradeoff between data size and quality from various camera specs. We only allow MTurk master workers to do the task at the beginning and to increase the diversity of workers, we accept the workers who have a 97% HIT acceptance rate and at least $10,000$ approved tasks. We manually checked summaries from Step 6 for quality assurance while filtering further based on empirically determined thresholds of how much eye gazing data was within the LLM response textbox(es) and the ratio of characters the users actually viewed.

Overall, 83 workers picked 275 topics out of 300 topics and complete 641 pointwise tasks and 695 pairwise tasks. 39 workers were identified as being below either of the thresholds (see Figure 28 in appendix). Further manual analysis of weak or non-committal summaries leads to the removal of 24 workers and 9.4% of tasks from the data collection.

4 User Behavior Analyses

We analyze how users read LLM responses using the gaze and mouse trajectories in IFLLM. In the pairwise setting, users see two responses side by side, which we refer to as the left and right response; in the pointwise setting, they see a single response. Throughout, we report behavior over normalized time, a rescaling of each session's timestamps to $[0, 1]$ using linear interpolation, which allows sessions of different absolute duration to be compared on a common axis.

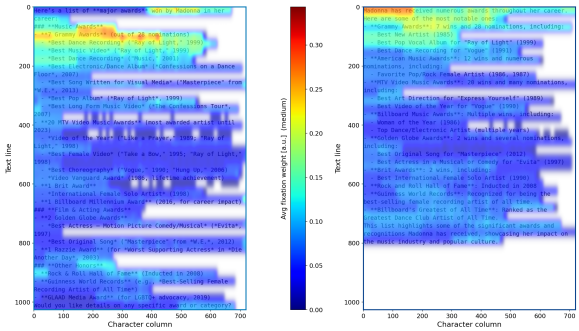


Figure 3: Average fixation weight over the response text in the pairwise setting, aggregated across all medium-length responses. The displayed text is a randomly selected example.

4.1 Aggregate Reading Patterns

On average, users give more attention to the early part of a response than to the rest. As shown in Figure 3, on the right response users concentrate on the opening words; on the left, attention shifts to the end of the first line and the start of the following lines, settling on the top-middle rather than the opening words that are commonly assumed to matter most.The same content receives attention in different places depending on its position, which suggests that the layout of the interface changes where users direct their attention.

Figure 4 indicates the average reading trajectory depends heavily on response length. Given a short response, users reach the end quickly and revisit text they have already read. As the response grows longer, they instead spend more time on the early portion of the response and progress more slowly. The same length split also governs how closely the mouse follows the gaze. For short responses, Figure 5 shows that the two are only weakly correlated. For medium and long responses, they are strongly correlated, as the user must move the mouse to the text box to scroll the longer response.

Figure 6 suggests that reading speed further depends on the task layout and the position of the response. Users read through the left response faster than the right. With only one response to read in the pointwise setting, users finish it early and have time to return to parts they have already seen.

4.2 Individual Variability

The aggregate patterns above describe the average user, but individual trajectories are highly irregular. For example, randomly selected trajectories in Figure 7 are full of back-and-forth movement, and they differ sharply from one another.

To understand the different types of patterns, we cluster gaze trajectories with BisectingK-Means (Steinbach et al., 2000) and visualize the centers in Figure 8. The clusters in Group 1 correspond to users who read the response and pausing to annotate or type the next query, though some reading quickly and others slowly. Group 2 captures the remaining styles: some users read only up to a point and then move back to what they have seen, some read at a steady rate through to the end, and some barely read the response at all.

5 Preference Prediction

To simplify our description and analysis, we focus on the implicit feedback collected for the side-by-side pairwise response comparison and mention the pointwise setting, which predicts the preference between current and previous responses, as an extension. We will first extract features from the implicit feedback and train a reward model to predict users' preference.

5.1 Feature Extraction

For every 0.1 second, we record their mouse position and gaze position from WebGazer. If their gazing point is inside a text box with a response, we record the character index they gaze at and the corresponding time. We then extract features from these trajectories to train our reward model. Table 1 summarizes all features; Mouse and gaze trajectories share the same file format, so mouse features mirror gaze features unless noted otherwise. In the pointwise setting, the left/right features become the current/previous features.

<u>TEXT FEATURES:</u> Basic properties of the query and responses, including query length and response length. Singhal et al. (2024); Dubois et al. (2024) show longer responses often receive higher scores.

<u>GAZE/MOUSE FEATURES:</u> When the users like a response, they tend to spend more time reading it in full (Yang et al., 2025b), so we summarize the trajectories into features of reading time and position. A one-second smoothing window is applied to time-based features to reduce noise.

5.2 Unused Features

After adding the implicit feedback features above, we find that the following features are either unused by the random forest or degrade its performance, so

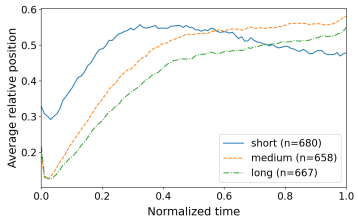


Figure 4: Average relative gaze position over normalized time, grouped by response length (short, medium, and long responses).

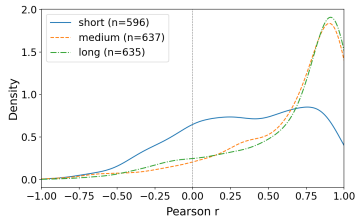


Figure 5: Distribution of the per-session Pearson correlation between mouse and gaze position, grouped by response length.

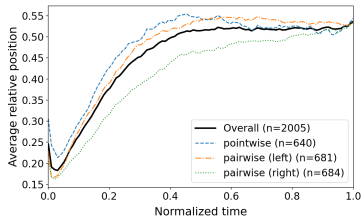


Figure 6: Comparison of average gaze trajectories from pointwise setting and left and right responses in the pairwise setting.

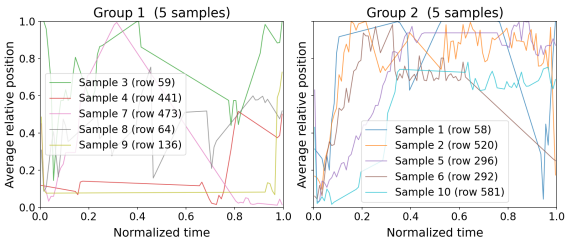


Figure 7: Gaze trajectories of ten randomly sampled sessions over normalized time.

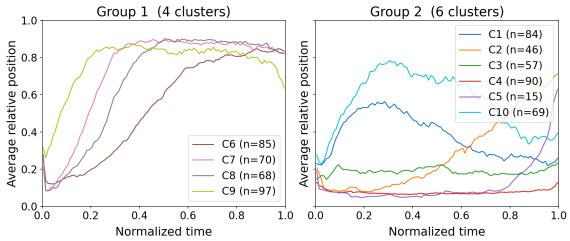


Figure 8: Gaze trajectory clusters over normalized time. The similar cluster centers are shown in the left figure (group 1).

we exclude them from our final model: LLM identity (one-hot indicators for the response source), bullet point prompt (whether the prompt included a bullet point instruction), and user identity (one-hot features for the top five most active users).

5.3 Reward Model Training and Analyses

To generate high-quality chat data, we select widely-used LLMs to generate responses, which usually do not have obvious errors for the users who are not familiar with the topic. This makes their preferences hard to predict. Table 2 shows that the zero-shot performances of Claude Sonnet 4.6 and Gemma-4 31B are close to 0.5, the level of random guesses. The supervised learning without implicit feedback also leads to similar performances. In Table 2, all the other methods conduct 5-fold cross-validation on 695 pairwise queries. For the standard reward models that take only the query and responses as the feature, the accuracy could only reach around 0.55 regardless of the size of the reward models.

To identify the useful features from implicit feedback, we first train a random forest (RF) on all features described in Section 5.1 of all data and keep only the top 50 features with the highest weights as our important feature (IF). We surprisingly find that the implicit feedback overpowers many features we considered effective, such as the identity of which LLMs generate the response and which user labels the preference. In both pairwise and pointwise settings, **RF + IF** achieves the best results.

To know the importance of gazing and mouse signal, we also train the random forest without mouse data and without gazing data (i.e., IF - Mouse/Gaze). Compared to **RF + (IF - Gaze)**, the worse performance **RF + (IF - Mouse)** in Table 2 suggests that removing mouse data is more detrimental than removing gazing data.

To understand why the mouse feature is so effective, we train three random forests in data that only have short, medium, and long responses and compare the total weights of the features of each signal source in Figure 9. The results show that random forest relies much more on the gazing data when the response is short. This suggests that some effectiveness of mouse signal comes from users' scrolling need because they might not point the mouse to the short responses they are reading. The complex interactions between the response length and implicit feedback features also justify our usage of random forest.[1] In Figure 9, we also observe that the response lengths, which are often the most important features in the standard reward models (Singhal et al., 2024), have much smaller weights than the implicit feedback.

[1]We also tried logistic regression which underperforms random forest.

Group	Feature	Description
Text	Query Length	Number of query characters
	Left/Right Response Length	Character count of each response
Gaze	Left/Right Max Character	Maximum character index read; serves as effective response length
	Left/Right Norm. Max Character	Max Character divided by Response Length
	Left/Right Total Records	Total number of gazing records ($\approx$ total seconds divided by 10)
	Left/Right Total/Reviewing Points	Gazing response points before/after excluding between-review periods
	Left/Right Total/Reviewing Norm. Points	Number of total/reviewing points divided by total records
	Left/Right Reviewing Time	Gazing response time during review
	Left/Right Reviewing Norm. Time	Reviewing gazing response time divided by total reviewing time
	Left/Right Avg/Var Norm. Character	Mean or variance of gaze character divided by Response Length
	Left/Right Avg Character in a Window	Mean gaze character in each of 20 equal time windows
	Proper Head Position Ratio	Fraction of time user's head is in the WebGazer-suggested green box
	Max Character Pairwise Comparison	$+1$ if left Max Character $>$ right, -1 otherwise
	Reviewing (Norm.) Time Diff	Left $-$ right reviewing (norm.) time
	Reviewing (Norm.) Time Ratio	Left divided by right reviewing (norm.) time
Mouse	Features identical to gaze features except for the head position feature	
Gaze and Mouse	Left/Right Ratio of Gaze and Mouse	Per-side Reviewing Time (Gaze) divided by Reviewing Time (Mouse)

Table 1: Feature descriptions for the reward models. Many gazing features have two versions. *Total* *: over all records and *Reviewing* * excluding estimated periods when the user annotates preferences or types a new query.

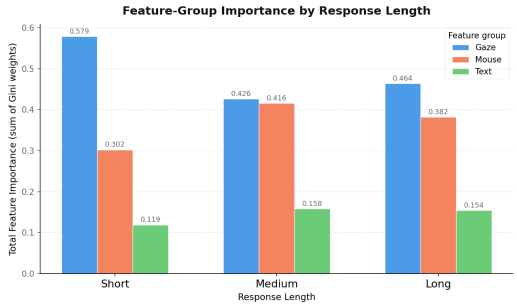


Figure 9: The comparison of features weights given different response lengths

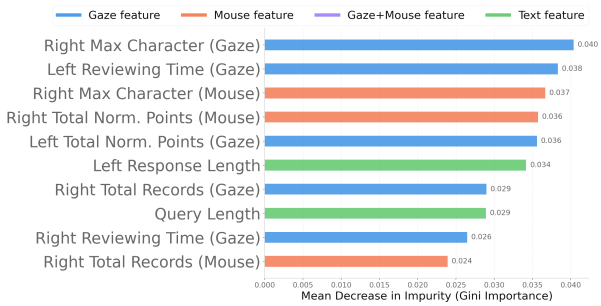


Figure 10: The importance weights of the top 10 features for our random forest model

Figure 10 visualizes the top 10 feature importance for **RF + IF** model. We can see that mouse and gazing both play important roles, and they are more important than the text length features, which are usually the strongest signal in the standard reward model (Singhal et al., 2024; Dubois et al., 2024). The various types of time features are ranked high because the users tend to spend more time on the response they like.

To analyze the feature influences on the prediction, we run the partial dependency analysis (Friedman, 2001), which plots the preference prediction changes by only varying the value of a feature on average across every sample. Figure 12 shows that the higher *Max Character*, the more likely they prefer the response because the users who like a response tend to finish reading it. However, the effect tends to saturate when the user only reads a little or has read a lot.

6 LLM Alignment

Section 5 demonstrates that implicit feedback, especially mouse movement, could drastically improve the accuracy of reward models predicting human preference. The next research question we investigate in this section is whether better reward models in the pairwise setting could be translated into better LLM alignment outcomes.

6.1 Training

For each reward model, we collect the predictions of 5 validation sets from the 5-fold cross-validation on the pairwise data. The 20% of these predictions are used as validation data. Our experiments test eight 1-4B base models, including GPT2-XL (Radford et al., 2019), Pythia 2.8B (Biderman et al., 2023), OLMo2 1B (Walsh et al., 2025), Llama3.2 3B (Grattafiori et al., 2024), Qwen2.5 1.5B, Qwen2.5 3b (Hui et al., 2024), Qwen3 1.7b, and Qwen3 4B (Yang et al., 2025a). Each of the base LLM is first supervisedly fine-tuned (SFT)

Method	Accuracy	F1 Class 1	F1 Class 0
Pairwise (695 Samples)			
Claude-S-4-6 + Zero-Shot	0.5303	0.5163	0.5434
Gemma-4-31B + Zero-Shot	0.5499	0.5252	0.5722
mBERT base + Text	$0.5549_{\pm 0.0347}$	$0.5220_{\pm 0.0366}$	$0.5815_{\pm 0.0353}$
mBERT base + Text + IF	$0.5896_{\pm 0.0193}$	$0.4733_{\pm 0.1194}$	$0.6319_{\pm 0.0172}$
mBERT large + Text	$0.5577_{\pm 0.0168}$	$0.5308_{\pm 0.0194}$	$0.5810_{\pm 0.0164}$
Qwen3 1.7B + Text	$0.5548_{\pm 0.0261}$	$0.5101_{\pm 0.0338}$	$0.5906_{\pm 0.0231}$
RF + All Features	$0.6074_{\pm 0.0073}$	$0.5776_{\pm 0.0132}$	$0.6322_{\pm 0.0070}$
RF + (IF - Gaze)	$0.6267_{\pm 0.0097}$	$0.5935_{\pm 0.0173}$	$0.6548_{\pm 0.0121}$
RF + (IF - Mouse)	$0.6089_{\pm 0.0130}$	$0.5836_{\pm 0.0162}$	$0.6313_{\pm 0.0157}$
RF + IF	$\mathbf{0.6415}_{\pm 0.0097}$	$\mathbf{0.6084}_{\pm 0.0173}$	$\mathbf{0.6694}_{\pm 0.0121}$
Pointwise (248 Samples)			
mBERT base + Text	$0.5239_{\pm 0.0667}$	$0.4716_{\pm 0.1042}$	$0.5636_{\pm 0.0495}$
mBERT base + Text + IF	$0.5319_{\pm 0.0434}$	$0.3325_{\pm 0.1903}$	$0.6254_{\pm 0.0648}$
RF + (IF - Gaze)	$0.6088_{\pm 0.0379}$	$0.6240_{\pm 0.0432}$	$0.5846_{\pm 0.0406}$
RF + (IF - Mouse)	$0.6006_{\pm 0.0185}$	$0.5935_{\pm 0.0252}$	$0.6044_{\pm 0.0183}$
RF + IF	$\mathbf{0.6291}_{\pm 0.0262}$	$\mathbf{0.6377}_{\pm 0.0301}$	$\mathbf{0.6137}_{\pm 0.0297}$

Table 2: Reward model performance comparison (average ± standard error across folds). IF refers to the important features from the implicit feedbacks. RF refers to random forest. Claude-S-4-6 means Claude Sonnet 4.6. mBERT means ModernBERT.

Reward Source	DPO/SFT Win Rate	DPO - SFT	Length
Explicit Feedback	0.4950 / 0.4625 ± 0.0127 / ± 0.0161	0.1079 ± 0.0529	208.8 ± 2.5
mBERT base + Text	0.4942 / 0.4554 ± 0.0086 / ± 0.0088	0.1221 ± 0.0542	227.3 ± 2.5
mBERT base + Text + IF	0.5071 / 0.4500 ± 0.0073 / ± 0.0082	0.1958 ± 0.0538	221.1 ± 2.5
RF + (IF - Gaze)	**0.5354 / 0.4196** ± 0.0119 / ± 0.0113	**0.3450** ± 0.0531	202.5 ± 2.6
RF + (IF - Mouse)	0.4871 / 0.4667 ± 0.0113 / ± 0.0107	0.1108 ± 0.0541	203.5 ± 2.5
RF + IF	0.5008 / 0.4592 ± 0.0139 / ± 0.0131	0.1892 ± 0.0538	206.6 ± 2.5

Table 3: Average response quality of 8 LLMs after DPO using different reward models. The quality is judged by GPT4.1-mini and averaged across 2400 prompts. Higher DPO winning rate and lower SFT winning rate is better. DPO - SFT means their average overall score difference. Explicit Feedback uses DPO + NLL, while other methods use rDPO + NLL. The standard errors are provided as our confidence region.

on all LLMs' responses. Next, we choose to use DPO that maximizes the probability of chosen data while minimizing the rejected response probability instead of conducting reinforcement learning (Ouyang et al., 2022), which is more expensive and often sensitive to hyperparameters, To further stabilize our experiments and avoid overfitting, we use rDPO (Chowdhury et al., 2024) with negative log likelihood (NLL) (Pang et al., 2024) for one epoch. Compared to the standard DPO, rDPO could emphasize the responses that are very confidently chosen by the reward model and NLL means adding an SFT loss on the chosen response. The **Explicit Feedback** from workers (i.e., the preference annotation) does not have a confidence, so we use DPO + NLL instead.

6.2 Testing

We randomly choose 300 pointwise queries for testing. GPT4.1-mini compares the responses from LLM after SFT and from LLM after (r)DPO + NLL and outputs the overall scores for each response from 1 to 10. They are tied for the same score. Otherwise, label DPO or SFT wins. In Appendix A.2, we also conduct human experiment to validate the scores from the LLM as a judge.

The results in Table 3 show that the preference predictions from **mBERT base + Text** only slightly improve the output response quality after rDPO + NLL, while **RF + (IF - Gaze)** achieves much better results with average 202.5 response length, which is significantly shorter than 228.6, the average length of SFT responses. Table 6 shows that the improvement is especially obvious for the recent models such as Qwen series.

The similar performances of **Explicit Feedback**, **mBERT base + Text**, and **RF + (IF - Mouse)** highlight the importance of the mouse movement signal. The unsatisfactory performances of **Explicit Feedback** show the importance of leveraging the confidence in rDPO. We also report the average performances of short, medium, and long response separately in Table 5. The improvement gap of **RF + (IF - Gaze)** steadily increases as the length of responses increases, while **RF + IF** seems to overfit the gazing data noise and degrade LLMs' capability for generating longer responses.

7 Conclusion

We introduced IFLLM, a dataset pairing webcam-based eye-gaze trajectories and mouse movements with explicit preference annotations. The dataset allows us to systematically measure the value of implicit feedback from users for the first time. The users exhibit complicated reading patterns, which are influenced by response length, interface layout, and individual style. Driven by the scrolling need for the long responses, users' mouse movement trajectories carry strong preference signal that text or even eye-gazing data cannot capture and drastically improve the accuracy of reward models and response quality from the resulting aligned LLMs. The effectiveness and accessibility of the mouse movement suggest a natural path toward a self-reinforcing data flywheel driven by ordinary user interactions.

Ethical Considerations

To collect eye-gazing and mouse trajectories data, we follow the protocol in our institution to acquire the IRB approval. We do not record video and all MTurk worker ID are anonymized before we release the data.

Our research might bring some positive impacts such as improving the factuality evaluation (Wanner et al., 2025) by emphasizing the parts the users might pay more attention to. In contrast, our research might encourage more companies to track user's mouse trajectories or even eye movements without users' consent, which might infringe users' privacy. Besides, data flywheel might reduce the diversity of possible LLM choices in the future.

Limitations

One limitation is that our reward model requires the implicit feedback as the input, which means at each round of RLHF (Ouyang et al., 2022), we need to show the responses generated by LLMs to the users to collect the required implicit feedback.

Due to the page limit, we haven't analyzed some the data we collected such as the likert score for each response and answers for post-QA questionnaires. To simplify our experiments, we also split the multi-turn question-answering into multiple single-turn question-answering sessions and leave the usage of cross-session context and signals as our future work.

References

James Allan, Eunsol Choi, Daniel Lopresti, and Hamed Zamani. 2024. Future of information retrieval research in the age of generative ai. *Computing Research Association (CRA)*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, and 1 others. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International conference on machine learning*, pages 2397–2430. PMLR.

Anna Bondar, David Robert Reich, and Lena Ann Jäger. 2025a. Aleyegnment: Leveraging eye-tracking-while-reading to align language models with human preferences. In *Proceedings of the First International Workshop on Gaze Data and Natural Language Processing*, pages 58–70.

Anna Bondar, David Robert Reich, and Lena Ann Jäger. 2025b. Colagaze: A corpus of eye movements for linguistic acceptability. In *Proceedings of the 2025 Symposium on Eye Tracking Research and Applications*, pages 1–9.

Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. 2024. Provably robust dpo: Aligning language models with noisy feedback. In *International Conference on Machine Learning*, pages 42258–42274. PMLR.

Shuwen Deng, Paul Prasse, David Reich, Tobias Scheffer, and Lena Jäger. 2024. Fine-tuning pre-trained language models with gaze supervision. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 217–224.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Ge Gao, Alexey Taymanov, Eduardo Salinas, Paul Mineiro, and Dipendra Misra. 2024. Aligning llm agents by learning latent preference from user edits. *Advances in neural information processing systems*, 37:136873–136896.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. In *Neural Information Processing Systems*. Curran Associates.

Eric Han, Jun Chen, Karthik Abinav Sankararaman, Xiaoliang Peng, Tengyu Xu, Eryk Helenowski, Kaiyan Peng, Mrinal Kumar, Sinong Wang, Han Fang, and 1 others. 2025. Reinforcement learning from user feedback. *arXiv preprint arXiv:2505.14946*.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, and 1 others. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 133–142, New York, NY, USA. Association for Computing Machinery.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and 1 others. 2024. Rlaif vs. rlhf: scaling reinforcement learning from human feedback with ai feedback. In *Proceedings of the 41st International Conference on Machine Learning*, pages 26874–26901.

Angela Lopez-Cardona, Sebastian Idesis, Miguel Barreda-Ángeles, Sergi Abadal, and Ioannis Arapakis. 2025a. Oasst-etc dataset: alignment signals from eye-tracking analysis of llm responses. *Proceedings of the ACM on Human-Computer Interaction*, 9(3):1–29.

Angela Lopez-Cardona, Carlos Segura, Alexandros Karatzoglou, Sergi Abadal, and Ioannis Arapakis. 2025b. Seeing eye to ai: Human alignment via gaze-based response rewards for large language models. In *The Thirteenth International Conference on Learning Representations*.

Sandeep Mathias, Diptesh Kanojia, Abhijit Mishra, and Pushpak Bhattacharya. 2020. A survey on using gaze behaviour for natural language processing. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 4907–4913. International Joint Conferences on Artificial Intelligence Organization.

Douglas W Oard and Jinliang Kim. 1998. Implicit feedback for recommender systems. In *AAAI Workshop on Recommender Systems*, pages 81–85.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Richard Y Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. 2024. Iterative reasoning preference optimization. *Advances in Neural Information Processing Systems*, 37:116617–116637.

Nikolas Papadopoulos. 2025. Eye-tracking as implicit feedback for aligning large language models and enhancing human-ai teaming. In *Proceedings of the 2025 Symposium on Eye Tracking Research and Applications*, pages 1–3.

Nikolas Papadopoulos, Shreenithi Navaneethan, Sheng Bai, Ankur Samanta, and Paul Sajda. 2026. Gaze patterns predict preference and confidence in pairwise ai image evaluation. *arXiv preprint arXiv:2603.24849*.

Alexandra Papoutsaki. 2015. Scalable webcam eye tracking by learning from user interactions. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, pages 219–222.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and 1 others. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. LaMP: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392, Bangkok, Thailand. Association for Computational Linguistics.

Andreas Säuberli, Darja Jepifanova, Diego Frassinelli, and Barbara Plank. 2026. Controlling reading ease with gaze-guided text generation. *arXiv preprint arXiv:2601.17781*.

Taiwei Shi, Zhuoer Wang, Longqi Yang, Ying-Chun Lin, Zexue He, Mengting Wan, Pei Zhou, Sujay Kumar Jauhar, Xiaofeng Xu, Xia Song, and 1 others. Wildfeedback: Aligning llms with in-situ user interactions and feedback. In *NeurIPS 2024 Workshop on Behavioral Machine Learning*.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2024. A long way to go: Investigating length correlations in rlhf. In *First Conference on Language Modeling*.

Michael Steinbach, George Karypis, and Vipin Kumar. 2000. A comparison of document clustering techniques.

Ningzhi Tang, Junwen An, Meng Chen, Aakash Bansal, Yu Huang, Collin McMillan, and Toby Jia-Jun Li. 2024a. Codegrits: A research toolkit for developer behavior and eye tracking in ide. In *Proceedings of the 2024 ieee/acm 46th international conference on software engineering: Companion proceedings*, pages 119–123.

Ningzhi Tang, Meng Chen, Zheng Ning, Aakash Bansal, Yu Huang, Collin McMillan, and Toby Jia-Jun Li. 2024b. Developer behaviors in validating and repairing llm-generated code using ide and eye tracking. In *2024 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 40–46. IEEE.

Evan Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, and 1 others. 2025. 2 olmo 2 furious (colm's version). In *Second Conference on Language Modeling*.

Hao Wang, Haocheng Yang, Licheng Pan, Lei Shen, Xiaoxi Li, Yinuo Wang, Zhichao Chen, Yuan Lu, Haoxuan Li, and Zhouchen Lin. 2026. Implicitrm: Unbiased reward modeling from implicit preference data for llm alignment. *arXiv preprint arXiv:2603.23184*.

Jianling Wang, Yifan Liu, Yinghao Sun, Xuejian Ma, Yueqi Wang, He Ma, Zhengyang Su, Minmin Chen, Mingyan Gao, Onkar Dalal, and 1 others. 2025a. User feedback alignment for llm-powered exploration in large-scale recommendation systems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 996–1003.

Yifan Wang, Bolian Li, Junlin Wu, Zhaoxuan Tan, Zheli Liu, Ruqi Zhang, Ananth Grama, and Qingkai Zeng. 2025b. Drift: Learning from abundant user dissatisfaction in real-world preference learning. *arXiv preprint arXiv:2510.02341*.

Miriam Wanner, Leif Azzopardi, Paul Thomas, Soham Dan, Benjamin Van Durme, and Nick Craswell. 2025. All claims are equal, but some claims are more equal than others: Importance-sensitive factuality evaluation of llm generations. *arXiv preprint arXiv:2510.07083*.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhongheng Yang, Aijia Sun, Yushang Zhao, Yinuo Yang, Dannier Li, and Chengrui Zhou. 2025b. Rlhf fine-tuning of llms for alignment with implicit user feedback in conversational recommenders. In *2025 4th International Conference on Artificial Intelligence, Internet of Things and Cloud Computing Technology (AIoTC)*, pages 587–591. IEEE.

Yifan Zhang, Chen Huang, Yueke Zhang, Jiahao Zhang, Toby Jia-Jun Li, Collin McMillan, Kevin Leach, and Yu Huang. 2025. Eyemulator: Improving code language models by mimicking human visual attention. *arXiv preprint arXiv:2508.16771*.

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

Qian Zhao, F. Maxwell Harper, Gediminas Adomavicius, and Joseph A. Konstan. 2018. Explicit or implicit feedback? engagement or satisfaction? a field experiment on machine-learning-based recommender systems. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, SAC '18, page 1331–1340, New York, NY, USA. Association for Computing Machinery.

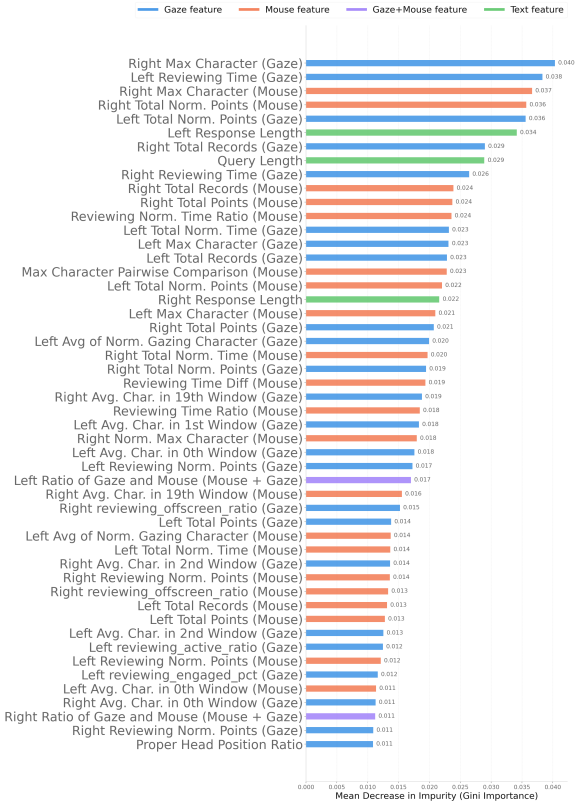


Figure 11: The importance weights of the top 50 features for our random forest model

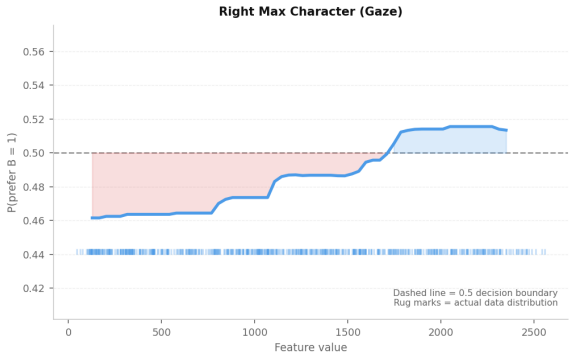


Figure 12: Partial dependency analysis on the last character index the user gazes at the right response. As the user reads right response further, the likelihood of the right response preference increases.

A More results

A.1 Preference Prediction

In Figure 11, we show the importance weights of all 50 features. We can see that most **Average of Characters in a Time Window** is pruned except the beginning and the end.

	Overall Human		Overall LLM	
Reward Source	**DPO-SFT**	**H vs H**	**DPO-SFT**	**H vs LLM**
mBERT base + Text	-0.407	0.351	0.470	0.355
RF + IF	0.383	0.252	0.470	0.495
	Comparison Human		Comparison LLM	
Reward Source	**DPO/SFT**	**H vs H**	**DPO/SFT**	**H vs LLM**
mBERT base + Text	0.288 / 0.475	0.388	0.400 / 0.533	0.233
RF + IF	0.500 / 0.367	0.352	0.567 / 0.400	0.277

Table 4: Comparison of human annotation and LLM as a judge. The responses from LLama3.2 3B for the first 30 prompts are judged. H vs H and H vs LLM mean Spearman correlation coefficient between the annotations from one MTurk worker to another worker or to LLM, respectively.

A.2 LLM Alignment

Although LLM as a Judge usually provides evaluations that are well correlated with human judgments (Lee et al., 2024), we spent $270 on MTurk for a small scale human experiment to further verify this in our setting. To facilitate effective factuality assessment, we asked MTurk workers to search the Internet because it's challenging to spot trivial errors in Llama3.2 3B's responses in the first glimpse. Each response is annotated by two master workers.

In Table 4, both human and GPT4.1 mini think **RF + IF** is significantly better than **mBERT base + Text**. The long responses often make the quality judgment difficult and subjective because different annotators might like different parts of the responses. The Spearman correlations between a worker and GPT4.1 mini are similar to the inter-annotator agreement, which validates the effectiveness of our LLM-as-a-judge evaluation results.

We present the average DPO performances given different response lengths in Table 5, which uses the response length from each LLM after DPO to group the responses into short, medium, and long. The results show that mouse signals are more useful for longer responses. Table 6 shows the performances of each LLM separately. We can see that the implicit feedback boosts the performance of most LLMs more. Finally, the average performance in each human evaluation dimension are reported in Table 7, which shows that **RF + IF** make the responses more relevant and factual while being slightly less informative.

A.3 Additional User Behavior Analyses

A.3.1 Example of Trajectories

We visualize the gaze trajectories from 5-turns QA in a task from one worker in Figure 13 and the corresponding mouse trajectories in Figure 14. We

Reward Source	Short		Medium		Long	
	DPO/SFT Score	DPO/SFT Win Rate	DPO/SFT Score	DPO/SFT Win Rate	DPO/SFT Score	DPO/SFT Win Rate
Explicit Feedback	**4.513**/4.385	0.494/0.456	4.944/4.909	0.485/0.476	5.456/**5.295**	0.506/0.455
ModernBERT + Text	4.447/4.379	0.469/0.474	4.856/4.830	0.481/0.468	5.621/5.350	0.532/0.425
ModernBERT + Text + IF	4.594/4.346	**0.520/0.439**	5.001/4.921	0.477/0.476	5.599/5.339	0.524/0.435
RF + (IF - Gaze)	4.635/4.417	0.481/0.469	**5.104/4.719**	**0.549/0.405**	**5.820**/5.388	**0.576/0.385**
RF + (IF - Mouse)	4.492/4.420	0.475/0.481	4.939/4.947	0.466/0.486	5.577/5.309	0.520/0.432
RF + IF	4.484/**4.280**	0.501/0.463	5.013/4.880	0.486/0.477	5.599/5.367	0.515/0.438

Table 5: Average quality of the responses with different lengths for 8 LLMs after DPO using different reward models.

Reward Source	GPT2-XL		Pythia 2.8B		OLMo2 1B		Llama3.2 3B	
	Overall	Win rate	Overall	Win rate	Overall	Win rate	Overall	Win rate
Explicit Feedback	3.09 / 3.01	0.483 / 0.433	3.92 / **3.69**	**0.520** / 0.430	4.44 / 4.62	0.440 / 0.530	6.20 / 6.07	0.510 / 0.460
ModernBERT + Text	**3.21 / 2.88**	0.487 / 0.440	3.93 / 3.85	0.513 / **0.427**	4.42 / 4.36	0.487 / 0.477	**6.48 / 5.94**	**0.543 / 0.430**
ModernBERT + Text + IF	3.16 / 2.90	0.513 / 0.420	3.98 / 3.91	0.497 / 0.453	4.59 / 4.49	0.503 / 0.473	6.39 / 6.18	0.533 / 0.450
RF + (IF - Gaze)	3.08 / 2.91	**0.540 / 0.393**	**4.05** / 3.88	0.500 / 0.447	**4.91 / 4.23**	**0.593 / 0.373**	6.43 / 6.17	0.507 / 0.467
RF + (IF - Mouse)	3.04 / 3.05	0.467 / 0.453	3.73 / 3.73	0.467 / 0.470	4.79 / 4.50	0.497 / 0.457	6.20 / 6.12	0.517 / 0.477
RF + IF	3.02 / 3.08	0.463 / 0.480	3.91 / 3.90	0.453 / 0.513	4.56 / 4.31	0.530 / 0.440	6.37 / 5.99	0.527 / 0.460
Reward Source	Qwen2.5 1.5B		Qwen2.5 3B		Qwen3 1.7B		Qwen3 4B	
	Overall	Win rate	Overall	Win rate	Overall	Win rate	Overall	Win rate
Explicit Feedback	4.99 / 4.60	0.547 / 0.403	5.97 / 5.85	0.497 / 0.470	4.84 / **4.66**	0.513 / 0.443	6.31 / 6.39	0.447 / 0.530
ModernBERT + Text	4.84 / 4.65	0.497 / 0.447	5.72 / 5.91	0.467 / 0.493	4.79 / 4.79	0.473 / 0.450	6.41 / 6.45	0.487 / 0.480
ModernBERT + Text + IF	5.00 / 4.61	0.530 / 0.430	6.04 / 5.76	0.527 / 0.433	4.90 / 4.82	0.467 / 0.477	6.45 / **6.26**	0.487 / 0.463
RF + (IF - Gaze)	**5.11** / 4.65	0.550 / 0.403	**6.24** / 5.89	**0.563 / 0.410**	**5.20** / 4.71	**0.533 / 0.413**	6.48 / 6.28	0.497 / 0.450
RF + (IF - Mouse)	4.90 / 4.55	0.533 / 0.417	6.10 / 5.91	0.513 / 0.450	4.96 / 4.85	0.453 / 0.493	6.30 / 6.44	0.450 / 0.517
RF + IF	4.95 / **4.45**	**0.560 / 0.400**	6.01 / **5.81**	0.510 / 0.440	4.91 / 4.93	0.457 / 0.500	**6.51** / 6.28	**0.507 / 0.440**

Table 6: Response quality of each LLM after DPO using different reward models. The quality is averaged across 300 prompts. The maximal DPO and minimal SFT are highlighted.

can see that the user could demonstrate diverse gaze behavior within a QA session.

A.3.2 Heatmaps for Long and Short Responses

Figures 15 and 16 extend the heatmap of Figure 3 to long and short responses. The length-dependent pattern from Section 4.1 holds: attention stays concentrated on the early portion of long responses, while concentration weakens as responses get shorter.

A.3.3 Mouse Reading Trajectories

The length effect from Section 4.1 also holds for the mouse trajectory (Figure 17). Grouped by response length, the mouse position over time follows the same pattern as the gaze trajectory in Figure 4: short responses are traversed quickly and then revisited, while for longer responses the mouse stays on the early portion and advances more gradually.

Grouped by task setting (Figure 18), the mouse trajectory follows the gaze trajectory in Figure 6: the pointwise setting advances fastest, and the left response is read faster than the right.

A.3.4 Position Distributions

The reading position can also be viewed spatially as the distribution of attention across the response. Grouped by response length (Figures 19 and 20), attention concentrates on the early portion of medium and long responses, while for short responses a large share of it falls at the end. Grouped by task setting (Figures 21 and 22), the distributions are largely similar across the pointwise and pairwise conditions, with attention concentrated near the start and end of the response in all three.

A.3.5 Gaze–Mouse Correlation

The mouse and gaze trajectories are positively correlated. Across all queries (Figure 23), the per-query correlation is positive on average, and computing it per user (Figure 24) shows that nearly every user follows this pattern. Grouped by task setting (Figure 25), the correlation is similar for the pointwise setting and for the left and right responses in the pairwise setting.

B Preference Prediction Details

B.1 Feature Extraction

Our gaze of an QA session are stored in a file. Since one session contains multiple queries, we need to preprocess the file to know each gaze record corresponds to which query. As mentioned before, we refer to one run as a "task" with one topic. During the task, we see that Step 5 of Figure 2

	RF + IF				mBERT base + Text			
Metrics	**DPO/SFT Win Rate**	**DPO-SFT**	**H vs H**	**H vs LLM**	**DPO/SFT Win Rate**	**DPO-SFT**	**H vs H**	**H vs LLM**
Factual	0.283 / 0.167	0.183	0.330	-	0.237 / 0.305	-0.153	0.148	-
Informativeness	0.283 / 0.333	-0.183	0.115	-	0.237 / 0.254	0.051	0.164	-
Relevancy	0.317 / 0.150	0.267*	0.301	-	0.203 / 0.220	-0.034	0.221	-
Overall	0.367 / 0.150	0.383*	0.252	0.495	0.254 / 0.322	-0.051	0.351	0.355
Comparison	0.500 / 0.367	0.333	0.352	0.277	0.288 / 0.475	-0.407	0.388	0.233

Table 7: Comparison of human annotation and LLM as a judge. H vs H and H vs LLM mean Spearman correlation coefficient between the annotations from one MTurk worker to another worker or to LLM, respectively. * means p<0.05

with "QA and Preference Annotation" is where our task-relevant eye and mouse tracking occurs (which we will refer to as "user data"). As mentioned before, about every 0.1 seconds we track the viewed character index, a short text span which includes that index (which we will refer to as the "viewed substring"), and gaze and mouse coordinates. This data is logged on a per-user, per-task basis. To connect each row of the user data with the associated query we process the rows in time sequential order. Query IDs are positive unique identifiers for each task query provided to the users. If the user is not looking at the screen, the webcam captures this, fills the relevant user data row with placeholder values and their query ID is inherited from the last matched task query (or -2 before any matches have occurred). If the user is looking at the screen, then we apply a character-windowed approach to determine what they are looking at during a particular time-step. We check whether the viewed substring appears in the source text within 15 characters before and after the tracked character index. If matched with the experiment instruction prompt we have provided to guide users, then we assign this a query ID of -1. If still unmatched, then we iterate the relevant set of task queries (either pairwise or pointwise) for said user to match the viewed substring with the relevant task query ID. On-screen data that didn't match any of the prior conditions are provided a default query ID of 0.

When computing the ratio of two features A and B, we use $\min(A/(0.001 + B), 100)$ to prevent from having a large value for small B. One second smoothing means that whenever we observe a gaze point side a response textbox, we assume the user still looks at that response in the next second to reduce the noise in the gazing data. Besides reviewing features, we also apply the one second smoothing is also applied to *Total Norm. Time*.

B.2 LLM Reward Model

We use the following prompt for Gemma 4 31B and Claude Sonnet 4.6 to get their zero-shot preference prediction.

```
You are an expert evaluator assessing
    the quality of two AI-generated
    responses to a user query.
Your task is to determine which response
     better answers the user's query.
Output your judgment as JSON with
    exactly two fields:
- "prediction": 1 if Response 1 is
    better, 2 if Response 2 is better
- "confidence": a float between 0.0 and
    1.0 indicating how confident you are
     (0.5 = completely uncertain, 1.0 =
    completely certain)
Output only valid JSON, nothing else.

User Query:
{query}

Response 1:
{response_1}

Response 2:
{response_2}

Which response better answers the user's
     query? Output JSON only.
```

There are 3 samples out of 695 samples that are not able to processed by Gemma 4 31B, so we ignore them when computing the performances. When we add the important features to the ModernBERT, we simply append every feature name and its value to the text of user query and the two responses.

B.3 Pointwise Settings

We also train our random forest reward model on the pointwise data, where it predicts the likert score of a single response [1-5]. We use the same text features and the implicit feedback features extracted from the mouse and gaze trajectories described in Section 5.1. The R^2 of the model is only around 0.05 under 5-fold cross-validation.

The score is difficult to predict because each worker might have different bias toward higher or lower scores and we also notice that workers rarely

Figure 13: An example of gazing trajectory for a topic

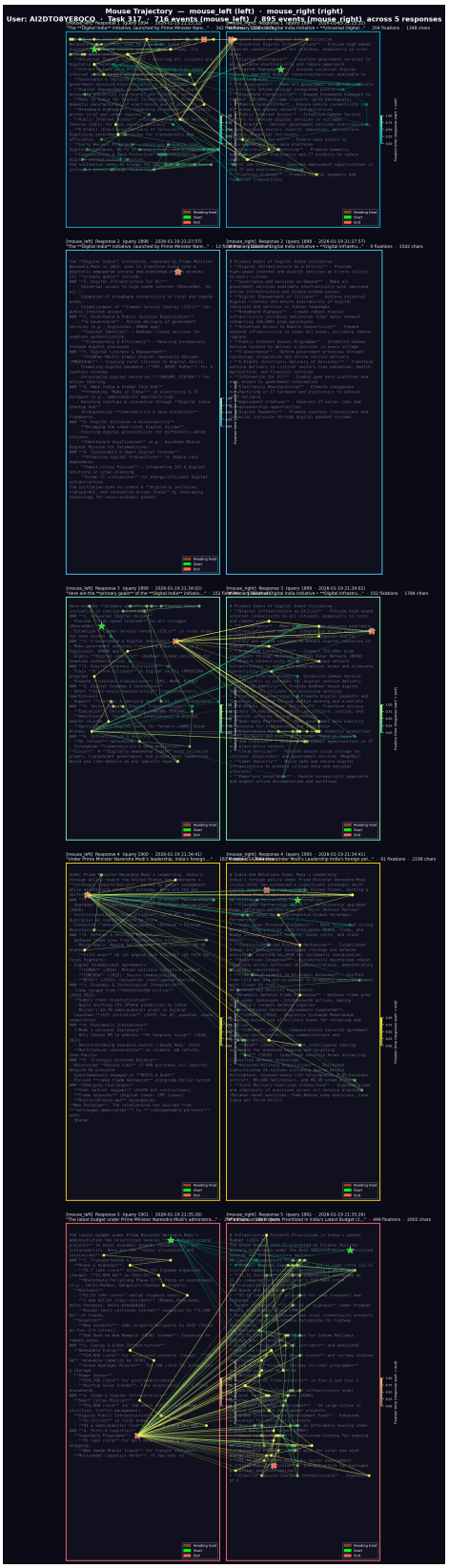


Figure 14: An example of mouse trajectory for a topic

give different score in a session. To force them to express their preference, we add the question of comparing with the previous question in the pointwise setting.

We discover that the workers have a strong bias: 70% of annotations prefer the current response

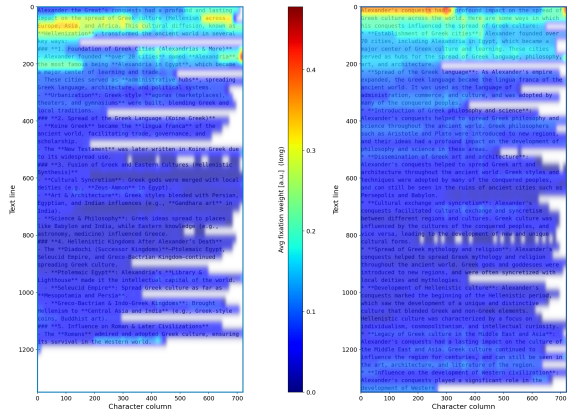


Figure 15: Average fixation weight over the response text in the pairwise setting, aggregated across all long responses. The displayed text is a randomly selected example.

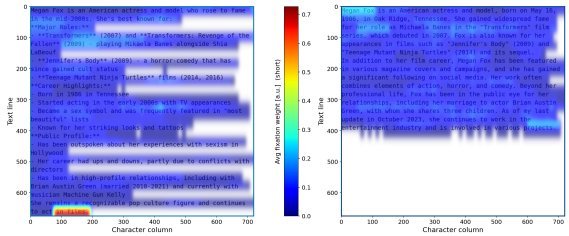


Figure 16: Average fixation weight over the response text in the pairwise setting, aggregated across all short responses. The displayed text is a randomly selected example.

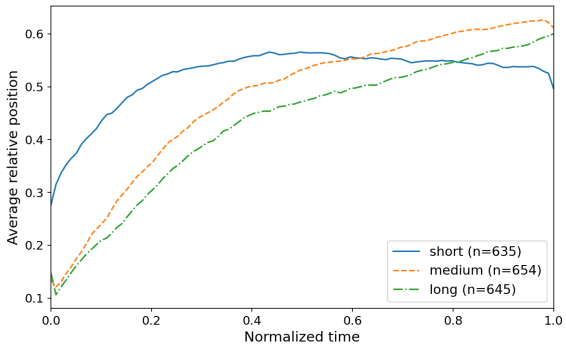


Figure 17: Average mouse position over normalized time, grouped by response length.

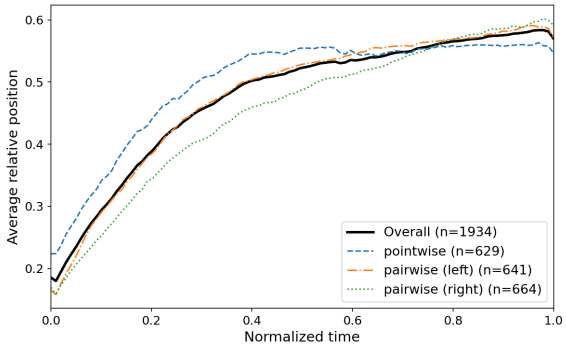


Figure 18: Average mouse position over normalized time, for the pointwise setting and for the left and right responses in the pairwise setting.

compared to the previous response. To balance the prediction classes, we subsample the data that prefer the current response.

B.4 Hyperparameters for ModernBERT and Random forest

For ModernBERT and Qwen3 1.7B, we set the batch size to be 1 and learning rate to be 1e-5. For pairwise, the number of epoch is 10 and for pointwise, which has fewer samples, we set the number of epoch as 5 to reduce overfitting.

We use the random forest implementation from Scikit-learn library (Pedregosa et al., 2011). When identifying the feature weights, we set max depth as 5 to capture more complex interaction and set the number of estimators as 200, minimal split as 10, and minimal leaf size as 4 to reduce overfitting. For the random forest that uses the important features, we use 5 max depth, 100 estimators, 5 minimal split, and 2 minimal leaf size.

We coarsely tune the hyperparameters of ModernBERT and random forest according to our validation scores, but we found that the performances are not sensitive to these hyperparameters.

C LLM Alignment Details

We modify the DPO implementation from `https://github.com/eric-mitchell/direct-preference-optimization` and use their default hyperparameter $\beta = 0.1$ and learning rate is $5e-7$. To reduce the memory requirement, we set batch size to be 2. All the models are trained using NVIDIA A100 80G.

We find DPO or rDPO along often decreases the loss by reducing the probability of both chosen and rejected probabilities, but reduce the rejected responses more. Adding the NLL/SFT term solves this problem.

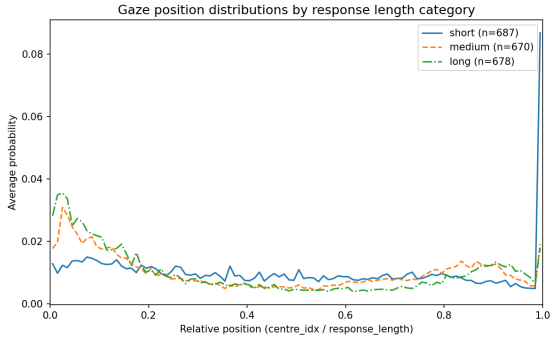


Figure 19: Gaze position distribution across the response, grouped by response length.

Figure 20: Mouse position distribution across the response, grouped by response length.

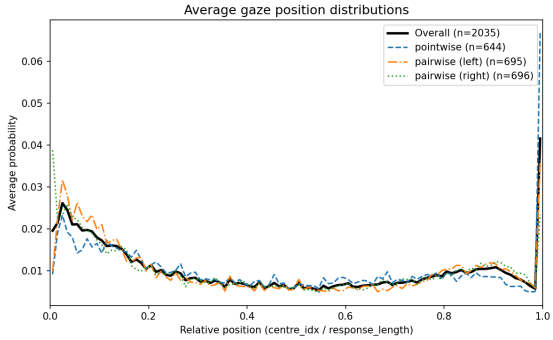


Figure 21: Gaze position distribution across the response, for the pointwise setting and for the left and right responses in the pairwise setting.

Figure 22: Mouse position distribution across the response, for the pointwise setting and for the left and right responses in the pairwise setting.

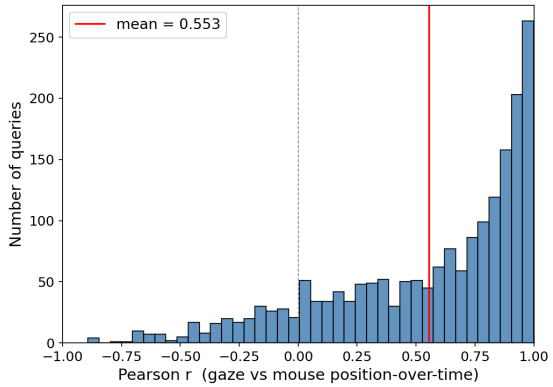


Figure 23: Distribution of the per-query Pearson correlation between mouse and gaze position over normalized time.

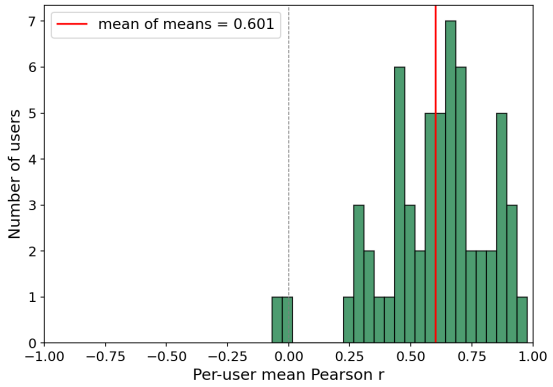


Figure 24: Distribution of the per-user mean Pearson correlation between mouse and gaze position over normalized time.

The prompt of LLM as a judge is listed below:

```
You are an expert evaluator assessing
    the quality of AI assistant
    responses.
You will be given a conversation prompt
    and two responses (A and B) from
    different AI models.

Evaluate each response on these criteria
    :
1. Instruction Following: Did the model
    follow all explicit and implicit
    instructions?
2. Informativeness: Is the response
    comprehensive without being verbose?
3. Factuality: Are the claims accurate?
    For creative prompts, judge internal
     consistency.
4. Clarity and Coherence: Is the
    response well-structured and easy to
     read?
5. Overall Helpfulness: Which response
```

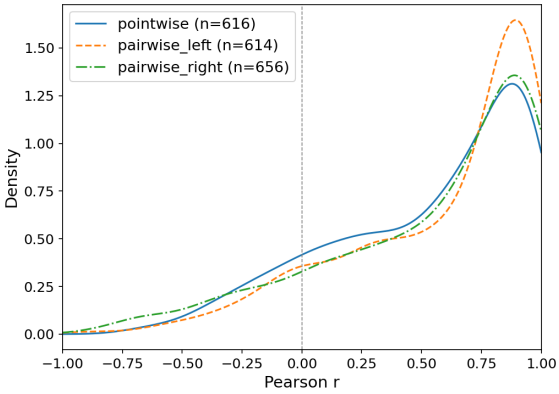


Figure 25: Distribution of the per-session Pearson correlation between mouse and gaze position, for the pointwise setting and for the left and right responses in the pairwise setting.

```
    is more ready to use for the human?

You MUST always respond in EXACTLY this
    format (no extra text, no markdown,
    no blank response):
SCORE_A: <integer 1-10>
SCORE_B: <integer 1-10>
WINNER: <A or B or tie>
REASONING: <one concise sentence>

Study these examples carefully before
    evaluating:

EXAMPLE 1
## Conversation Prompt
Human: What is the capital of France?

## Response A
The capital of France is Paris. It has
    been the country's political and
    cultural centre for centuries.

## Response B
France.

SCORE_A: 9
SCORE_B: 3
WINNER: A
REASONING: Response A directly and
    accurately answers the question with
     useful context, while Response B
    names the country instead of its
    capital.
---
EXAMPLE 2
## Conversation Prompt
Human: Write a short poem about autumn.

## Response A
Leaves fall like whispered secrets,
Gold and red adorn the trees,
Crisp air carries distant echoes
Of summer's last, reluctant breeze.

## Response B
Autumn is a season. Trees lose leaves.
    It gets cold.

SCORE_A: 9
SCORE_B: 2
WINNER: A
REASONING: Response A fulfils the
    creative request with imagery and
    rhythm; Response B is a flat,
    prosaic description with no poetic
    quality.
---
EXAMPLE 3
## Conversation Prompt
Human: How do I reverse a list in Python
    ?

## Response A
You can reverse a list in Python using
    the built-in reverse() method:
    my_list.reverse() modifies it in
    place, or use my_list[::-1] to get a
     new reversed list.

## Response B
Use the reverse function on the list
    object. It will reverse the list for
     you.

SCORE_A: 8
SCORE_B: 5
WINNER: A
REASONING: Response A provides two
    concrete, correct methods with brief
     code examples, while Response B is
    vague and offers no actionable
    syntax.
---
Now evaluate the following pair using
    the EXACT same format as the
    examples above.
```

C.1 LLM Alignment Human Experiments

Our MTurk template could be seen in Figure 26. Only master workers could do the task. We provide $1.6 or $2 wage for each task, which takes around 10 minutes. We choose to test Llama3.2 3B because it could output coherent responses with some errors for workers to find. 9 out of 120 responses are rejected by Claude code and manual inspection.

Our website allows the users to ask follow-up questions, so some queries are ambiguous without showing the previous queries. We instruct the MTurk workers to allow the LLMs to interpret query freely (e.g., *What is her most important role?* does not mention who *she* refers to, so the responses from LLMs can talk about any actress).

D AI Usage

We use Claude code to generate some analysis codes and MTurk Template. We also use Claude, Gemini, and ChatGPT to help us develop the website, search for some related work, or provide writ-

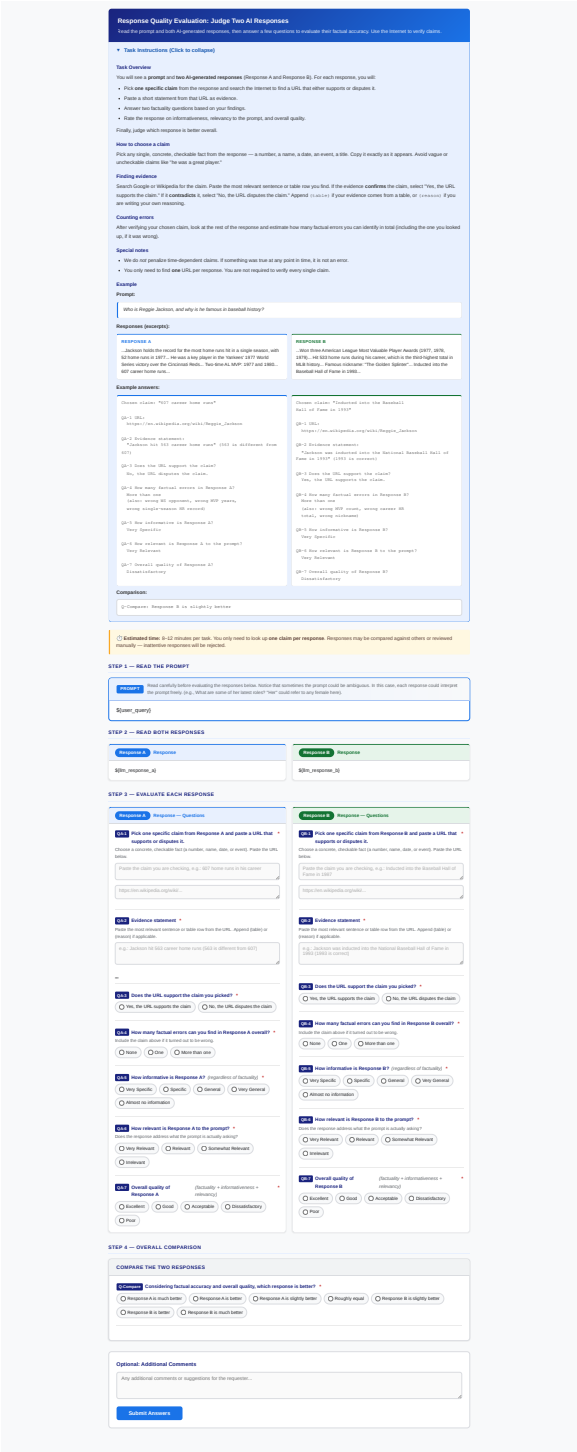


Figure 26: The crowdsourcing template we used in our LLM alignment experiment.

ing suggestions.

E Website Details

Our website is developed using PHP and MySQL database. MTurk workers might use various types of browsers and often do multiple tasks in parallel. Each user is allowed the use of Google Chrome, Firefox, and Microsoft Edge. Multiple windows and tabs are supported but 1 tab total is encouraged.

E.1 Login and Pre-test Questionnaire

We manually filtered topics for content sensitivity or being too niche for non-factoid conversation such as "Jeffrey Epstein" and "Biggest ball of twine".

The General Information Questionnaire consists of 2 pages. The first page requests for user consent of the experiment and acknowledges the use of a web camera. Note the browser itself requests for camera use as well. The second page is a questionnaire on the background of the user such as demography and highest education level with an emphasis on flexibility. We mention in the consent page all data is secured in our server.

Both new and old users are met with the Instruction Page, Step 3. It contains a list of instructions that detail high quality queries with positive and negative examples, the webpages they can expect, and troubleshooting if the webcam does not work. In the instruction, we encourage workers to move the mouse to the places they gaze. The rest of the experiment features a navigation bar with a hyperlink to the Instruction Page to further its accessibility.

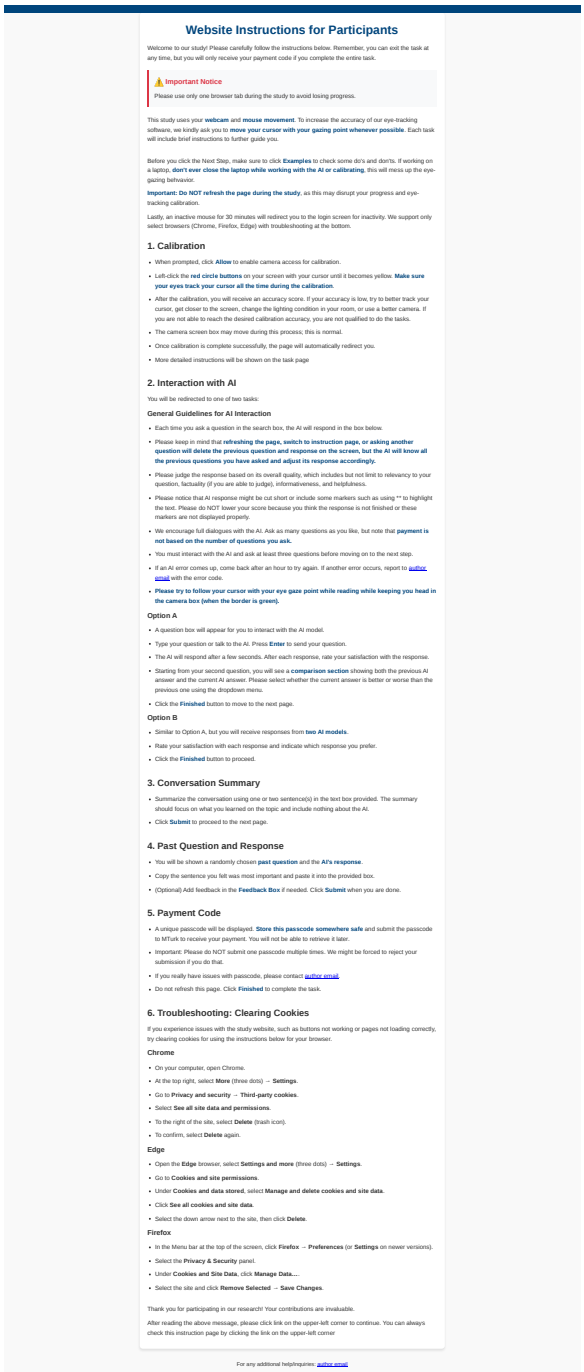
Any head position out of the green box may incur poor prediction. The calibration uses 8 buttons around the screen the user must move the mouse and click multiple times. Alongside the mouse is a red dot constantly displaying the prediction of Webgazer. The red dot allows the user to understand the prediction model but remains a distraction for further steps in the experiment, hence the red dot is only for calibration. The button presses assume the user is gazing at the button with each press, acting as a ground truth for the current eye gazing point. Afterwards, the user moves the mouse and their gaze to the center to measure accuracy. If a suitable accuracy is met (refer to 3.4, the user may proceed with the experiment.

E.2 QA and Preference Annotation

Both pointwise and pairwise contain a small instruction set at the top, a webcam, the query box, and the navigation bar. The full instruction could be seen at Figure 27.

E.3 Quality Control

The variables chosen in Figure 28 best represent user integrity and associated quality cutoffs (0.75



Website Instructions for Participants

Welcome to our study! Please carefully follow the instructions below. Remember, you can exit the task at any time, but you will only receive your payment code if you complete the entire task.

⚠ Important Notice

Please use only one browser tab during the study to avoid losing progress.

This study uses your **webcam** and **mouse movement**. To increase the accuracy of our eye-tracking software, we kindly ask you to **move your cursor with your gazing point whenever possible**. Each task will include brief instructions to further guide you.

Before you click the Next Step, make sure to click **Examples** to check some do's and don'ts. If working on a laptop, **don't ever close the laptop while working with the AI or calibrating**, this will mess up the eye-gazing behavior.

Important: Do NOT refresh the page during the study, as this may disrupt your progress and eye-tracking calibration.

Lastly, an inactive mouse for 30 minutes will redirect you to the login screen for inactivity. We support only select browsers (Chrome, Firefox, Edge) with troubleshooting at the bottom.

1. Calibration

- When prompted, click **Allow** to enable camera access for calibration.
- Left-click the **red circle buttons** on your screen with your cursor until it becomes yellow. **Make sure your eyes track your cursor all the time during the calibration**.
- After the calibration, you will receive an accuracy score. If your accuracy is low, try to better track your cursor, get closer to the screen, change the lighting condition in your room, or use a better camera. If you are not able to reach the desired calibration accuracy, you are not qualified to do the tasks.
- The camera screen box may move during this process; this is normal.
- Once calibration is complete successfully, the page will automatically redirect you.
- More detailed instructions will be shown on the task page

2. Interaction with AI

You will be redirected to one of two tasks:

General Guidelines for AI Interaction

- Each time you ask a question in the search box, the AI will respond in the box below.
- Please keep in mind that **refreshing the page, switch to instruction page, or asking another question will delete the previous question and response on the screen, but the AI will know all the previous questions you have asked and adjust its response accordingly.**
- Please judge the response based on its overall quality, which includes but not limit to relevancy to your question, factuality (if you are able to judge), informativeness, and helpfulness.
- Please notice that AI response might be cut short or include some markers such as using ** to highlight the text. Please do NOT lower your score because you think the response is not finished or these markers are not displayed properly.
- We encourage full dialogues with the AI. Ask as many questions as you like, but note that **payment is not based on the number of questions you ask.**
- You must interact with the AI and ask at least three questions before moving on to the next step.
- If an AI error comes up, come back after an hour to try again. If another error occurs, report to author email with the error code.
- **Please try to follow your cursor with your eye gaze point while reading while keeping you head in the camera box (when the border is green).**

Option A

- A question box will appear for you to interact with the AI model.
- Type your question or talk to the AI. Press **Enter** to send your question.
- The AI will respond after a few seconds. After each response, rate your satisfaction with the response.
- Starting from your second question, you will see a **comparison section** showing both the previous AI answer and the current AI answer. Please select whether the current answer is better or worse than the previous one using the dropdown menu.
- Click the **Finished** button to move to the next page.

Option B

- Similar to Option A, but you will receive responses from **two AI models**.
- Rate your satisfaction with each response and indicate which response you prefer.
- Click the **Finished** button to proceed.

3. Conversation Summary

- Summarize the conversation using one or two sentence(s) in the text box provided. The summary should focus on what you learned on the topic and include nothing about the AI.
- Click **Submit** to proceed to the next page.

4. Past Question and Response

- You will be shown a randomly chosen **past question** and the **AI's response**.
- Copy the sentence you felt was most important and paste it into the provided box.
- (Optional) Add feedback in the **Feedback Box** if needed. Click **Submit** when you are done.

5. Payment Code

- A unique passcode will be displayed. **Store this passcode somewhere safe** and submit the passcode to MTurk to receive your payment. You will not be able to retrieve it later.
- Important: Please do NOT submit one passcode multiple times. We might be forced to reject your submission if you do that.
- If you really have issues with passcode, please contact author email.
- Do not refresh this page. Click **Finished** to complete the task.

6. Troubleshooting: Clearing Cookies

If you experience issues with the study website, such as buttons not working or pages not loading correctly, try clearing cookies for using the instructions below for your browser.

Chrome

- On your computer, open Chrome.
- At the top right, select **More** (three dots) → **Settings**.
- Go to **Privacy and security** → **Third-party cookies**.
- Select **See all site data and permissions**.
- To the right of the site, select **Delete** (trash icon).
- To confirm, select **Delete** again.

Edge

- Open the **Edge** browser, select **Settings and more** (three dots) → **Settings**.
- Go to **Cookies and site permissions**.
- Under **Cookies and data stored**, select **Manage and delete cookies and site data**.
- Click **See all cookies and site data**.
- Select the down arrow next to the site, then click **Delete**.

Firefox

- In the Menu bar at the top of the screen, click **Firefox** → **Preferences** (or **Settings** on newer versions).
- Select the **Privacy & Security** panel.
- Under **Cookies and Site Data**, click **Manage Data...**.
- Select the site and click **Remove Selected** → **Save Changes**.

Thank you for participating in our research! Your contributions are invaluable.

After reading the above message, please click link on the upper-left corner to continue. You can always check this instruction page by clicking the link on the upper-left corner

For any additional help/inquiries: author email

Figure 27: Our website instruction page

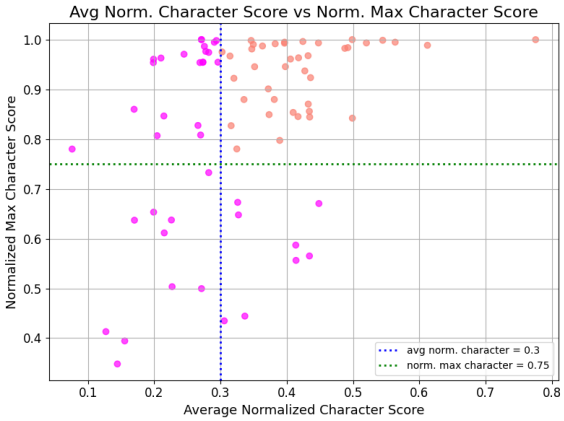


Figure 28: Macro average of each user's Average Normalized Character (i.e., Total Norm. Points) vs Norm. Max Character. Magenta points were users below thresholds, in consideration for removal of dataset

for response score and 0.3 for max index score) for the representation of the user's attention to the task.