Behavioral Indicators of Overreliance During Interaction with Conversational Language Models

Chang Liu*
c-liu21@tsinghua.org.cn
Tsinghua University
Beijing, China

Qinyi Zhou*
qinyi.zhou@connect.ust.hk
Hong Kong University of Science and Technology
Hongkong, China

Xinjie Shen
xinjie@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Xingyu Bruce Liu
xingyuliu@ucla.edu
UCLA
Los Angeles, California, USA

Tongshuang Wu
sherryw@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

Xiang 'Anthony' Chen
xac@ucla.edu
UCLA
Los Angeles, California, USA

Abstract

LLMs are now embedded in a wide range of everyday scenarios. However, their inherent hallucinations risk hiding misinformation in fluent responses, raising concerns about overreliance on AI. Detecting overreliance is challenging, as it often arises in complex, dynamic contexts and cannot be easily captured by post-hoc task outcomes. In this work, we aim to investigate how users' behavioral patterns correlate with overreliance. We collected interaction logs from 77 participants working with an LLM injected plausible misinformation across three real-world tasks and we assessed overreliance by whether participants detected and corrected these errors. By semantically encoding and clustering segments of user interactions, we identified five behavioral patterns linked to overreliance: users with low overreliance show careful task comprehension and fine-grained navigation; users with high overreliance show frequent copy-paste, skipping initial comprehension, repeated LLM references, coarse locating, and accepting misinformation despite hesitation. We discuss design implications for mitigation.

CCS Concepts

• **Human-centered computing** → **User studies**; **Web-based interaction**; **Empirical studies in HCI**.

Keywords

Overreliance on AI, Language Models, Interaction Behaviors, Human-AI Collaboration

ACM Reference Format:

Chang Liu, Qinyi Zhou, Xinjie Shen, Xingyu Bruce Liu, Tongshuang Wu, and Xiang 'Anthony' Chen. 2026. Behavioral Indicators of Overreliance During Interaction with Conversational Language Models . In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 23 pages. https://doi.org/10.1145/3772318.3790332

*Both authors contributed equally to this research.

CHI '26, April 13–17, 2026, Barcelona, Spain

ACM ISBN 979-8-4007-2278-3/2026/04
https://doi.org/10.1145/3772318.3790332

1 Introduction

Conversational LLMs, *e.g.*, ChatGPT [64], Gemini [30, 31], and Claude [73], are increasingly used for various open-ended tasks, from creative writing to problem-solving. These models hold promise for enhancing human productivity and creativity by providing knowledge-driven assistance. As a result, many users have come to rely on LLMs for reasons such as fast information access, reduced cognitive load, perceived authority, and lower social costs of seeking assistance. However, LLMs' responses, while often appearing fluent and plausible, may contain misinformation stemming from hallucinations, outdated information, and prompt injection attacks [4, 40, 105]. When users over-rely on LLMs and fail to critically evaluate their responses, such misinformation can lead to serious errors. Ultimately, this undermines the effectiveness of human-AI collaboration, reduces efficiency, and results in low-quality content [34, 68, 103].

In this paper, we focus specifically on overreliance on LLMs, which is defined as **"users accepting incorrect LLM recommendations"** [11, 42, 44, 69, 99]. Different from trust, which is an internal attitude that may lead to reliance [37, 102] , reliance itself is the action taken by the human that can be observed and measured. The conventional approach of understanding users' overreliance is through measuring and comparing the outcomes of tasks with *vs.* without AI assistance [5, 28, 43].

However, in the case of conversational LLMs, focusing solely on task outcomes overlooks the interaction process, which often spans multiple rounds of question-answering. While each round can be considered a task outcome in itself, evaluating individual rounds in open-ended tasks is difficult, as the content of each round may not directly relate to the final result. These intermediate rounds influence the final outcome, and errors caused by overreliance during these interactions are often hard to trace back to specific causes or points of failure in the final result. Therefore, in this work, we investigate an alternative approach — **how users' interaction behaviors[1] (mouse- and keyboard-based activities) correlate with their overreliance on the conversational LLMs**. By capturing such behaviors, we can go beyond post-hoc analyses of final outputs and instead enable just-in-time detection of overreliance

[1]We defined the interaction behavior as the patterns of action event sequence (e.g., mouse click, scrolling, keypress). We refer to action event sequence as "action sequences" for brevity.

during task execution. This, in turn, opens the door to adaptive mitigation strategies (*e.g.*, Rzeszotarski and Kittur [77]) that intervene only when necessary, without disrupting every user interaction.

Inspired by prior work that examines how crowdworkers' interaction patterns relate to task quality on a HIT page [27, 36, 79], we investigate whether interaction behaviors similarly correlated to different levels of overreliance on conversational LLMs. For example, imagine two users performing a writing task with ChatGPT. The user with high overreliance, after receiving the LLM's response, may superficially scan the content, copy and paste the whole text without carefully reviewing it. The user with low overreliance on the other hand, may review the LLM's response thoroughly, and only selectively copy-paste task-relevant segments of sentences.

To identify user behaviors linked to overreliance, we first collected interaction data from 77 participants, who completed three distinct tasks by collaborating with a conversational LLMs pre-injected with plausible misinformation. Critically, this misinformation was designed to simulate naturally occurring LLM errors (*e.g.*, hallucinations, outdated information, and prompt injection attacks), while maintaining full experimental control over its content and placement. To systematically measure user overreliance, we evaluated the extent to which participants' final task answers were influenced by the pre-injected misinformation. Existing researches [13, 47, 52, 100] have adopted this method as a balance between ecological plausibility and experimental reproducibility. To reflect real-world scenarios, the tasks were tied to three of the most representative and frequently used information sources for accuracy judgment. Throughout the interaction, we logged a comprehensive set of action events, *e.g.*, click, scrolling, keypress. To analyze the behavior data, we employed a sequence-aware, state-of-the-art clustering method [108]. To classify the behavioral patterns, we processed user action sequences, optimized clustering reliability, extracted typical behavioral patterns, and interpreted their cognitive implications.

As a result, our analyses identified five distinct behavioral patterns related to overreliance: During task execution, users with low overreliance tended to carefully read task details at the beginning and work more independently after verifying the LLM's response. They navigated the interface in a fine-grained, context-specific manner, as they often modified text after pasting, rather than copying and pasting blindly. In contrast, users with high overreliance frequently referred back to the LLM's responses throughout the task. Even when they initially hesitated on LLM's response and asked follow-up questions, they ultimately trusted and adopted it. Their navigation was coarser and visually guided, focusing on prominent or predictable interface areas-locations that were easy to locate—because they performed little editing and relied heavily on repeatedly copying entire sections at once. We interpret participants' cognition by mapping each behavioral pattern to the cognitive theories whose core features closely align with the behavioral characteristics, and we reinforce these interpretations by cross-validating them with participants' strategy reports.

Overall, we contribute a new understanding of overreliance on conversational LLMs by examining users' interaction behaviors. Specifically:

- We collected a dataset [2] linking users' LLM interaction behaviors with quantitative measurements of their overreliance, which can serve as an entry point in understanding the relationship between overreliance and behavioral patterns.
- We adapted and refined a cluster-based analytical framework, tailoring it to quantify the relationship between behavioral data and overreliance—thereby enhancing the robustness of behavioral-to-reliance inference.
- We identified five distinct behavioral patterns corresponding to overreliance, complemented by empirical evidence, cognitive interpretation, and design implications for mitigating overreliance.

2 Related Work

We first review existing frameworks for overreliance on LLMs and limitations of current mitigations. Then we introduce existing outcome-oriented overreliance metrics and the feasibility of inferring states from interaction behaviors—highlighting the gap our work fills: leveraging user behaviors to understand and characterize behavioral correlates of overreliance on conversational LLMs, as a first step toward real-time detection.

2.1 Overreliance on AI: Existing Frameworks and Unique Challenges of Conversational LLMs

Previous works have defined overreliance on AI as "users accepting incorrect AI recommendations" [69], "following its suggestions even when those suggestions are wrong and the person would have made a better choice on their own" [11].

Modern language models (*e.g.*, GPT [64] and Gemini [30, 31]) primarily rely on the transformer architecture [101] which learns from massive amounts of internet data to predict the next token (e.g, words in a sentence) based on how it is contextually related to the previous tokens. While next-token prediction has shown a general power of solving various problems—from question answering to creative writing, the underlying transformer architecture necessarily brings along significant "side effects". Generating unfaithful or misleading outputs (*i.e.*, hallucinations) is one of the key challenges in developing and using LLMs [6, 9, 14, 41, 109] while combating their "misinformation harms" [26, 51, 62, 104] and aligning with human values [15, 67, 85]. Specifically, misinformation harms include disseminating false or misleading information, causing material harm by disseminating false information (*e.g.*, in giving medical or legal advice), and leading users to perform unethical or illegal actions [38, 104].

Conversational LLMs's unique characteristics, such as non deterministic outputs that confuse users and complicate verification [2, 81], erroneous backtracking when challenged [49, 50], sensitivity to indirect input attributes (e.g., epistemic markers, sycophancy, sandbagging) [71, 86, 112], and fast generation of high-volume novel content that imposes cognitive burden, raises verification costs, and leads users to mistake fluency/length for accuracy [1, 95, 97], fostering the risk of overreliance on AI.

[2]https://github.com/CJunette/behavior_indicator_of_overreliance

Existing mitigations for overreliance on AI primarily include: explanations [25, 33, 83, 87], uncertainty expressions [3, 48, 90, 98], providing recommendations only upon request [29, 72], altering speed of interaction [66], and cognitive forcing functions [11, 17, 82]. However, these mitigations have double-edged effects: explanations risk backfiring [33, 87, 92], uncertainty expressions are plagued by poor model calibration [57, 74], and cognitive forcing functions risk under-reliance and impose additional cognitive burden [17, 70, 83]. These shortcomings highlight that current one-size-fits-all mitigation strategies fail to account for the nuanced human-AI interaction. It is necessary to develop adaptive mitigation *that leverages real-time, process-level signals (e.g., behavioral traces) during the interaction process* [5, 93].

2.2 From Outcome to Process-Oriented Understandings of overreliance on LLMs

Different from trust (i.e., a human attitude that may lead to reliance [37, 102]), reliance itself refers to observable human actions, which can be operationalized via specific overreliance metrics and quantified.

Current research on measuring users' overreliance on LLMs mainly relies on *outcome-oriented* metrics. Such studies infer overreliance levels from the final results of users' task completion, with key measures including "whether users adopt or agree with AI's correct/incorrect outputs" [42, 84, 94]. For example, users are defined as overreliant if they make errors due to trusting LLMs' hallucinations, or if they adopt AI suggestions at a high rate while disregarding independent judgment. However, this outcome-oriented approach has two critical limitations: First, it overlooks the *process* (i.e., how humans interact with AI on the interface) that leads to outcomes, failing to capture behavioral details during interaction. As a result, it cannot distinguish between "active rational reliance" and "passive blind compliance", hindering the design of adaptive mitigations for different kinds of overreliance users. Second, given the diversity of real-world LLM usage scenarios (e.g., academic writing, trip planning, programming), it is impractical to design scenario-specific outcome measures for every task, making it hard to establish a unified benchmark for overreliance on LLMs and undermining the generalizability of this approach.

While most research on measuring overreliance on LLMs focuses on outcome-oriented metrics, a smaller body has begun to explore other dimensions—specifically how users' interaction behaviors and contextual features correlate with overreliance on LLMs. For instance, CUPS [58] analyzes how programmers interact with code-recommendation systems, and proposes interaction inefficiencies and excessive time costs as potential behavioral signals of overreliance. Building on dual-process theories, prior work suggests that overreliance on LLMs often arises when users default to fast, intuitive, low-effort System 1 thinking instead of slower, effortful, analytical System 2 thinking [11, 20], and that such cognitive strategies can be reflected in behavioral patterns, such as how thoroughly users read, edit, or verify AI outputs. Metacognitive research [95] links AI overreliance to users' metacognitive challenges, which manifest as observable behaviors. REL-A.I. [111], an interaction-centered framework, explores contextual features (e.g., knowledge domain, perceived LM competence) as contextual signals that modulate overreliance on LLMs, and relates these features to observable interaction patterns. Swaroop et al. [93] investigated accuracy–time tradeoffs in AI-assisted decision making under time pressure, finding that overreliers and non-overreliers demonstrate differentiated use of AI assistance types, and that a user's overreliance rate serves as a key behavioral correlate. However, existing studies on process-oriented correlates of overreliance on LLMs remain fragmented: they propose isolated signals but lack a systematic and quantified framework that organizes interaction behaviors into behavioral patterns that correlate with overreliance on LLMs.

2.3 Interaction Behaviors as Performance Correlates

Human behaviors are widely recognized as important predictors of individuals' cognitive states and task outcomes, and user event logs—such as records of clicks, mouse movements, and keypresses—can effectively encode information about human interaction patterns and the subsequent task performance [12, 56, 108]. Rzeszotarski and Kittur [80] propose *task fingerprinting*—a means to capture the process that crowdworkers use to complete a task, consisting of their interactions with the task interface, such as clicks, scrolls, and keypresses. Crowdspace supports the evaluation of crowdwork by combining workers' behavioral traces with their task outputs through mixed-initiative machine learning, visualization, and interactive techniques [78]. Other related works use behavioral traces to pre-select workers' results [27] and predict the quality of web page structure annotation [36].

In the era of Generative AI, this behavioral-inferential logic remains valid: Ziegler et al. [113] identified a correlation between developer interactions and perceived productivity on GitHub through regression analysis, revealing that the acceptance rate of AI-generated suggestions is a more reliable predictor of productivity than alternative metrics. CoAuthor captures rich interactions between writers and GPT-3 in creative writing tasks and demonstrated AI as a writing "collaborator" [54].

All this prior work demonstrates the feasibility of mining users' interaction behaviors to obtain process-level signals that correlate with high-level performance-related measures, although none has employed such an approach to understand overreliance on LLMs. Our work addresses this gap by characterizing users' interaction behaviors that correlate with outcome-level overreliance, providing a behavioral basis for future real-time detection and adaptive mitigation.

3 Data Collection Study

To investigate how users exhibit overreliance on conversational LLMs during interaction, we conducted a controlled laboratory experiment (approved by our institute's IRB). To simulate diverse real-world use cases, we designed three task scenarios, each framed around a different information source. In every task, we deliberately injected plausible misinformation into the LLM's responses, simulating naturally occurring errors such as hallucinations, outdated knowledge or prompt injection. Users completed the tasks following a fixed procedure within a set time limit. Overreliance was defined as the extent to which users' final submissions incorporated

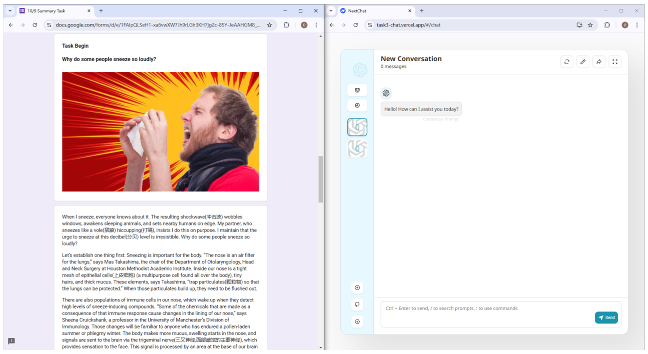


Figure 1: Interface setup in the experiment. Participants are asked to use a split-screen, the left half of the screen will display the task-related page, and the right half will display the LLM page as well as the page for search engine.

or were influenced by the injected misinformation. We collected detailed action logs of their interactions during the experiment. This setup allows us to systematically measure overreliance while also capturing the fine-grained behavioral patterns that accompany it.

3.1 Study Design

Participants. We recruited 77 participants from two local universities. Participants averaged 22.40 years old (SD=9.22), with 30 having high school/some college, 31 a Bachelor's, 10 a Master's, and 2 a Doctoral degree. On a 5-point Likert scale, participants' familiarity with conversational LLMs averaged 3.0 (SD = 1.2) out of 5, and participants' frequency of conversational LLMs usage averaged 3.8 (SD = 1.8) out of 5. The experiment took approximately 1.5 hours, and participants received $25 as compensation.

Procedure. Prior to the experiment, participants provided informed consent (Appendix A.4), completed a demographic and background survey, and were guided via a video tutorial to install a browser extension—which collected behavioral data throughout task completion. After each task, participants submitted the answer and uploaded behavioral logs from the extension. Upon completing all tasks, they filled out a questionnaire assessing LLM trust, task familiarity, and decision-making strategies (Section A.12). Technical assistance was available via online chat throughout the experiment. Total duration was approximately 1.5 hours.

Task Platform. We conducted the experiment online, where participants opened the experiment link in a Chrome web browser on their own PCs to participate. Based on the open-source project ChatGPT-Next-Web[3] and the GPT-4 model, we developed a privatized dialogue platform for our experiment, which serves as the conversational LLMs. Figure 1 demonstrates the setup participants use for the user study.

Behavioral Data Logging. We used Chrome extensions to collect user behavior data, via JavaScript scripts that utilize event listeners to monitor user actions on the web page, including *mouseMovement*, *clicks*, *scrolling*, *keypress*, *copy*, *paste*, *highlighting*, *delete*, and *idleness* (defined as 3 seconds of inactivity). Additionally, we monitored window switch behaviors on the LLM interface, which indicate that users are using alternative tools other than the LLM.

We logged the user actions on task page and LLM page separately into two files.

Please refer to Table 1 for details of all the actions and features we collected.

3.2 Task Design and Misinformation Injections

To determine the task design, we conducted expert interviews with LLM misinformation researchers to discuss and justify the guiding principles for task selection.

(1) The task should be commonly recognized as tasks where an LLM is typically used.
(2) The tasks cover three common sources to judge LLM-provided information: *(i)* judging by personal knowledge (common sense), *(ii)* judging by limited contextual information, and *(iii)* judging by external information, e.g., via online search.
(3) The tasks should have clear measurable outcomes, allowing a straightforward assessment of overreliance.

As a result, we chose three tasks—*quiz solving*, *article summarization*, and *trip planning*, which we describe in Table 2. Check Appendix A.1 for details about each task. *quiz solving* included two trials under two conditions (with *vs.* w/o LLM), yielding 3×2×2 = 12 different task orders, which we counterbalanced across participants.

Pilot studies were first used to determine the appropriate time allocation based on the difficulty of Task 1, ensuring that participants could complete the task with the assistance of the LLM. Using this time constraint as a reference, we subsequently calibrated the difficulty of Task 2 and Task 3. For Task 2, we selected an essay of moderate difficulty, and for Task 3, we adjusted the required number of submissions. These adjustments ensured that participants would need LLM assistance to complete the tasks within the allotted time. Furthermore, before beginning the tasks, participants were instructed to complete the task with the LLM and to engage seriously with the experiment; otherwise, their compensation might be reduced.

To systematically study overreliance, we injected task-specific plausible misinformation into LLM outputs. Following prior work [13, 47, 52, 100], this design simulates natural misinformation while enabling systematic evaluation of overreliance by measuring misinformation in participants' final outputs. In the rest of this section, we describe the detailed design for each task, including: 1) the rationale for choosing the task, 2) the nature of misinformation injected, and 3) the evaluation of overreliance, following the framework below.

3.2.1 *Quiz Solving.*

This task was selected to simulate real-world scenarios where users rely on LLMs for decision-making, contrasting behaviors when participants use only personal common sense versus LLM-provided information. Specifically, participants ranked the importance of survival items in two scenarios (survival on the moon and survival in the desert) and under two conditions (*with LLM* and *without LLM*). The four conditions were counterbalanced to mitigate learning effects.

[3]https://github.com/ChatGPTNextWeb/ChatGPT-Next-Web

Action	Action Description	Action Attribute	Action Feature Description
mouseMovement	When the mouse moves.	`total_mouse_movement`	The total distance of mouse movement.
		`mouse_movement_duration`	The duration of a mouse move.
click	When the mouse clicks.	-	-
scroll & mousewheel	Mouse wheel actions in scrolling. Scroll event triggers when an element has been scrolled.	`scroll_duration`	The duration of a scroll.
		`scroll_distance`	The distance of a scroll.
		`scroll_direction`	The direction of a scroll.
keypress	When the user presses keys on the keyboard.	`keypress_duration`	The duration of a keypress.
		`keypress_keyCount`	The count of inputs during a keypress.
copy	When the user copies text.	`copy_textLength`	The length of copied text.
paste	When the user pastes text.	`paste_textLength`	The length of pasted text.
highlight	When the user selects certain text.	`highlight_textLength`	The length of highlighted text.
delete	When the user deletes text using actions such as backspace, delete, or cut (Ctrl+X).	`delete_duration`	The duration of a deletion.
		`delete_keyCount`	The count of deletions.
idleness	When the user doesn't input any action for over 3 seconds.	`idle_duration`	The duration of an idleness.
* element switch	When the user conducts an element switch.	-	-
* tab switch	When the user switches to another tab within the LLM window.	-	-
* prompt input	When the user writes prompts within the LLM window.	-	-

Table 1: User Behaviors and Corresponding Features Collected for the Study. This table presents the preprocessed user behaviors and their associated attributes/features collected via Chrome extensions (in both LLM and Task pages) using JavaScript event listeners. The collected behaviors include both general web actions (e.g., mouse movement, clicks, scrolling, and keypresses) and LLM page-specific actions (marked with *: element switch, tab switch within the LLM window, and prompt input in the LLM window). For each behavior, the table details its description, action attributes, and feature descriptions.

Misinformation Injection. To mimic real-world LLM hallucinations while avoiding obvious absurdity, we altered the expert-validated ranking of survival items to generate plausible yet incorrect orderings. The LLM was instructed to consistently output this erroneous ranking and provide convincing but false justifications when participants requested explanations. **Evaluation Metrics.** We used the NASA standard scoring algorithm (Appendix A.3) to calculate a deviation score (S), where lower scores indicate closer alignment with the ground truth [21, 53, 63]. Overreliance was quantified as the difference in performance scores between the *with LLM* and *without LLM* conditions. The raw difference was normalized using the minimum and maximum scores across participants.

3.2.2 Article Summarization. This task reflects practical use cases of LLMs for text processing (e.g., literature abstracting, news summarization). It assesses participants' ability to evaluate whether LLMs' interpretations align with the content provided by users.

Misinformation Injection. According to previous methods [13, 47, 52, 100], we generate detail-tampered misinformation by instructing the LLM to slightly alter facts from the original text. Eleven misinformation instances requiring contextual reasoning to detect were manually selected and embedded in the LLM's system prompt; the LLM was instructed to persist with these errors even if challenged.

Evaluation Metrics. Normalizing the count of retained misinformation against the total number of injected instances. Overreliance is calculated as the occurrence of misinformation in users' summary answers.

3.2.3 Trip Planning. This task simulates real-world scenarios where users rely on LLMs for accessing time-sensitive information (e.g., API integration, trip planning)—contexts in which external search engines serve as the primary tool to verify such time-sensitive

Task	Description	Design	How LLM Generates Misinfo.	Overreliance Metrics
Quiz Solving	Rank the importance of 15 items in a survival scenario.	2 trials × {w/, w/o} LLM, 15 mins per trial	Manually create an incorrect ranking and ask the LLM to provide a reasonable explanation (e.g., "Matches are crucial for survival on the moon due to limited oxygen availability").	$\frac{(S_{\text{w/LLM}} - S_{\text{w/o LLM}}) - \min}{\max - \min}$
Article Summarization	Summarize a 700-word scientific article.	Single task, with LLM, 15 mins	Ask the LLM to generate a summary with errors (e.g., changing "sneezing is to flush out particulates" in the original text to "sneezing is to flush out cells").	$\frac{\text{Retained misinfo. count}}{\text{Total misinfo. count}}$
Trip Planning	Collect information for trip planning.	Single task, with LLM, 15 mins	Ask the LLM to generate suggestions with errors (e.g., generating locations that are not in the destination, or providing incorrect admission prices).	$\frac{\text{Retained misinfo. count}}{\text{Total misinfo. count}}$

Table 2: Details of the Three Tasks Designed for the Study. The table includes each task's identifier, description, design, misinformation injection method, and metric for measuring user overreliance on LLMs.

content. Specifically, participants were instructed to plan a trip to Copenhagen for a specific date.

Misinformation Injection. Similar to Article Summarization, we first asked the LLM to generate accurate travel recommendations for Copenhagen, then instructed it to modify these recommendations into fact-checkable misinformation. Twenty such verifiable misinformation instances were selected. The LLM was further instructed to provide these pieces of misinformation with plausible justifications whenever participants inquired about relevant topics.

Evaluation Metrics. Instances of misinformation retained in participants' itineraries. Given the task's openness, GPT-4 was used for automated detection via named entity recognition and logical inference (e.g., identifying false admission prices or fabricated attraction locations). The evaluation normalizes the count of retained misinformation against the total injected instances.

4 Analysis

In this section, we present the methods used to process and analyze the collected interaction data. The goal is to uncover behavioral patterns correlated with user overreliance on LLMs. We processed the interaction logs to enable meaningful behavioral analysis. Preprocessing ensured data completeness and interpretability. Vectorization converted events into a format suitable for computational analysis. Segmentation allowed examination of behavior at different temporal scales. Autoencoder embedding produced consistent low-dimensional representations of variable-length sequences. Clustering identified recurring behavioral patterns, and post-clustering selection retained only robust, informative clusters for further interpretation of users' cognitive and behavioral strategies.

Preprocessing. We first preprocessed the data for downstream analysis. Incomplete logs resulting from upload errors or user-interaction issues were removed. To reduce redundancy and improve interpretability, low-level events (e.g., mouse movements, scrolls, keystrokes) were merged into higher-level action events using temporal and semantic heuristics. All events were then labeled by page context (Task or LLM), merged by timestamp, temporally aligned, and normalized so that timestamps range from 0 to 1, representing the full session duration. Details of filtering, merging, and normalization are provided in Appendix A.13.1.

Vectorization. To ensure that log events could be consistently encoded and processed, we vectorized all event data. Each action event is encoded as a 37-dimensional feature vector representing four components: the action type (15 dimensions), timestamp (1 dimension), page type (2 dimensions), and various type-specific continuous and categorical attributes (19 dimensions). Table 6 details each feature and the number of dimensions allocated to it (Appendix A.13.2). These features cover all parameters recorded in the event log (i.e., the Action Attributes listed in Table 1), ensuring that our event encoding preserves the full information present in the original logs.

Segmentation. To analyze user behavior at varying temporal granularities, we segmented each session into fixed-duration windows ranging from 10 to 60 seconds, with a 1-second stride. The rationale behind this choice is that shorter windows yield isolated action events that lack meaningful patterns, whereas longer windows often produce highly repetitive sequences. These segments serve as the fundamental behavioral units for encoding and clustering. Segments inherit the overreliance label (one score per task per participant), allowing us to examine how localized behaviors relate to overall task performance.

Autoencoder Embedding. To produce consistent representations across variable-length action sequences, we used a transformer-based autoencoder that projects each sequence into a low-dimensional latent space while preserving essential behavioral information.

A special `[CLS]` token was prepended to each input sequence and served as a global summarization vector. The encoder maps the entire sequence into latent space, and the decoder seeks to reconstruct the original. The final embeddings used for clustering are taken from the `[CLS]` output vector of the encoder.

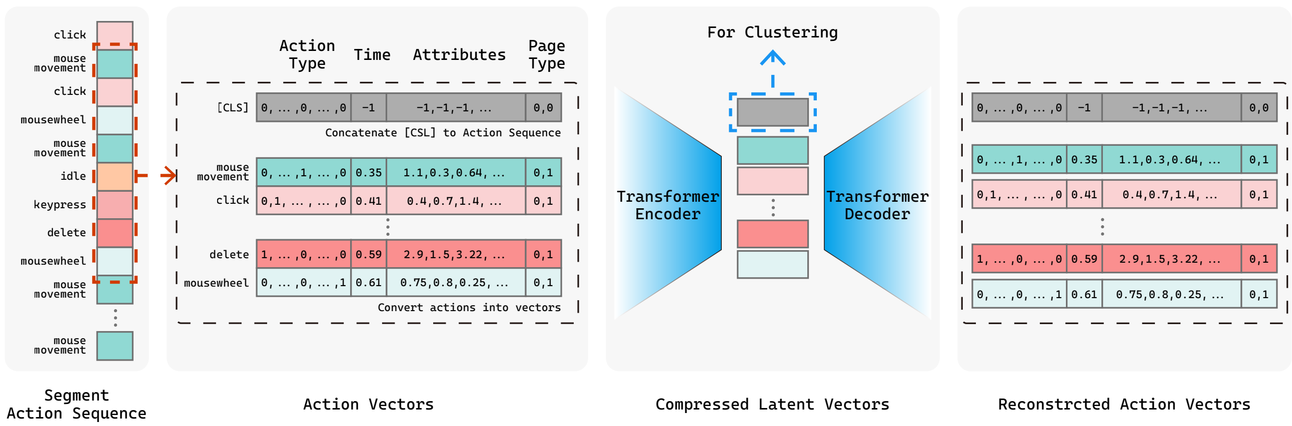


Figure 2: Overview of the Analysis Pipeline. We segment user interaction logs into overlapping time-based windows, encode each sequence into standardized feature vectors, use an autoencoder to produce compact sequence embeddings, and cluster these embeddings to identify recurring behavioral patterns. Selected clusters are interpreted in terms of user overreliance.

We trained a separate model for each combination of task and window size (3 tasks × 6 windows = 18 models). A full model specification and training loss breakdown are provided in Appendix A.13.3.

Clustering. We clustered the `[CLS]` latent vectors using DBSCAN [22]. To ensure robustness, we explored multiple clustering thresholds and neighborhood sizes (eps and min_samples), retaining only those clusters that appeared consistently across parameterizations.

For the test set, we assigned cluster labels using k-nearest neighbors (k = 5) in the embedding space. Clustering and assignment parameters are described in Appendix A.13.4.

Post-Clustering Selection. To retain only meaningful and generalizable patterns, we applied two criteria to filter clusters:

- **Intrinsic Similarity**: The distribution of overreliance scores in the training and test set assigned to a cluster must not significantly differ (two-tailed t-test, $p > 0.05$). It indicates that members within a cluster exhibit a similar distribution of overreliance.
- **Predictive Capability**: The average overreliance score of training members must accurately predict that of test members (details in Appendix A.13.5). It indicates that the cluster's average overreliance level reliably reflects its members' tendencies.

For each selected cluster, we extracted the 20 latent vectors nearest the cluster centroid and retrieved their corresponding action sequences. These representative sequences were manually analyzed to interpret behavioral and cognitive strategies exhibited by users under varying levels of overreliance. Overreliance levels were binned into high, neutral, and low categories for interpretive purposes as described in Appendix A.13.6.

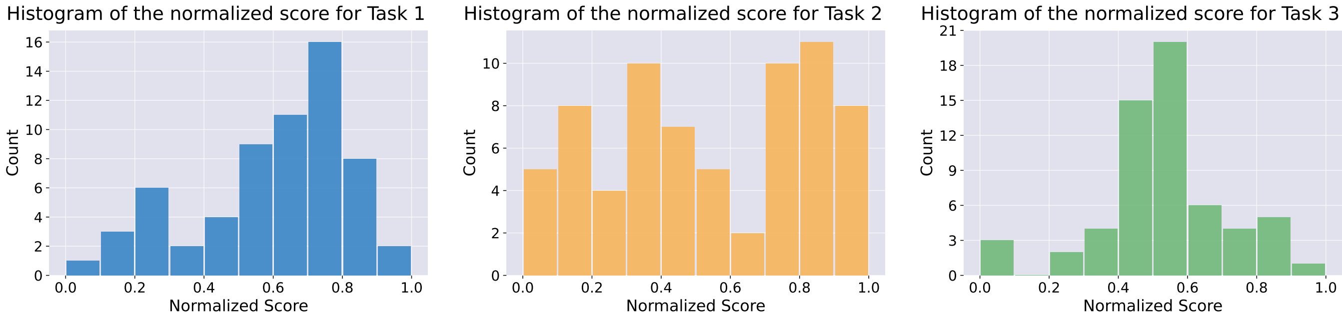
5 Findings

Figure 3 shows the histogram of normalized overreliance scores for three different tasks. A lower overreliance score indicates a lower degree of overreliance on AI, while a higher value signifies a greater level of overreliance.

As mentioned in Section A.3, it is essential to ensure no significant difference in $\text{score}_{\text{trial}}$ between the two trials of Task 1. Therefore, we conducted further validation on the collected data.

After clustering the 18 combinations, we identified a total of 154 distinct clusters. Applying the selection criteria from Section 4, we filtered these down to 54 core valid clusters with good predictive capability and strong overreliance similarity. Among these, 18 salient clusters have significantly high or low overreliance in the training set, and 36 clusters correspond to cases with no significant difference. The presence of these clusters suggests that our method can distinguish behavioral patterns associated with higher versus lower overreliance levels in our dataset. We also retrieved participants' answers to the decision-making strategies in post-task questionnaire after each task. We first linked each behavioral cluster to its corresponding participants and tasks, and then retrieved the post-task strategy reports for those cluster members. We coded these reports using a codebook aligned with our identified behavioral patterns and theoretical constructs related to overreliance on LLMs, iteratively refining codes and themes in team meetings and writing analytic memos to capture how self-reported strategies supported or challenged our behavioral interpretations.

In this section, we further analyze the 18 salient clusters associated with high or low overreliance, and examine consistently recurring action sequence patterns across the broader set of 54 core valid clusters. For each interaction pattern, we first introduce the ***Frequency of the Feature***, followed by ***Behavioral Characteristics***—including the action sequence (Action Sequence) and a behavioral interpretation based on participants' self-reported



(a) ***quiz solving*** (b) ***article summarization*** (c) ***trip planning***

Figure 3: Histograms of normalized overreliance scores for three distinct tasks, where each subfigure corresponds to one task, the vertical axis (*Count*) represents the number of participants falling into each score bin, and the horizontal axis (*Normalized Score*) represents the normalized measure of overreliance on AI. A lower normalized score indicates a lower degree of overreliance on AI, while a higher score signifies a greater level of overreliance. The three tasks include: (a) *quiz solving*, (b) *article summarization*, (c) *trip planning*.

Overreliance	Action Sequences, Interpreted Behavior, Cognitive Process
	(a) Frequency of Copying-Pasting Task: Trip planning, Window: 10–60s, $N = 32$
High	Repeated *copy-paste*, no text editing. Adopt LLM's responses directly.
Low	*Keypress/delete* before/after pasting, less *copy-paste*. Cautious copying + post-paste editing.
	(b) Focused Task Comprehension at the Start Task: Article summarization (Window: 10–50s, $N = 22$); Trip planning (Window: 10–50s, $N = 13$)
Low	Consecutive alternating *mousewheel + scroll* at task start. Carefully read task initially.
	(c) Frequency of Referring LLM's responses Task: Quiz solving, Window: 10s, $N = 25$
High	Alternated between LLM & Task pages, single *mousewheel + mouseMovement* on LLM then *click* on Task each time. Refer to LLM's responses per ranking.
Low	Scroll (*mousewheel*) on LLM page, then switched to Task page & do *mouseMovement + click* series. After referring to LLM's response, finish all the rankings independently.
	(d) Coarse- vs. Fine-Grained Locating & Editing Task: Article summarization, Window: 50–60s, $N = 34$
High	Consecutive *mousewheel/scroll + mouseMovement*, lengthy *copy-paste* at end. "Rough" navigation (prominent areas).
Low	Few *mousewheel/scroll* at start ⟶ *mouseMovement + click + keypress + delete*. "Refined" editing (specific areas).
	(e) Users with high overreliance: Pausing and Hesitation Before LLM Prompting Task: Quiz solving, Window: 50s, $N = 9$
High	Long *idle* on Task page ⟶ *keypress + delete* on LLM page. Initially hesitate, then trust & adopt LLM's responses.

Table 3: Summary of User Interaction Behaviors, Interpretations, and Cognitive Processes Associated with Different Levels of overreliance on LLMs.

decision-making strategies (Behavior). Then we make ***Interpretation*** of the possible underlying cognitive states.

For better illustration, we summarized the results of our quantitative analysis in Table 3, and visualized the simplified action sequence patterns in Figure 4; raw action sequences are provided in Appendix A.8.1. The implementation details of visualization is also included in Appendix. A.8.1.

5.1 Frequency of Copying-Pasting

Frequency of the Feature. This pattern frequently appears in the high overreliance clusters across window 10-60s for *trip planning*, which covers 36 participants.

Behavior Characteristics. Two distinct behavior sequences were identified (As shown in Figure 4.a). Action Sequence The copy-paste action consists of a *highlight* and *copy* on the Task/LLM page, a *mouseMovement* to the other page, and then a *paste* there.

Users with high overreliance: Behavior users usually perform 5–6 copy/paste actions within a 60s window. The copied content is often lengthy, usually consisting of an entire paragraph *Users with low overreliance*: Behavior users usually perform 1-3 copy/paste actions within a 60s window. The *paste* action is sometimes followed by *keypress* and *delete* action to revise the pasted content.

Interpretation. For users with high overreliance, the high copy-paste frequency reflects rapid, low-effort uptake of LLM's responses: users may bypass deliberate information screening and reviewing, aligning with System 1's core attributes-automaticity, minimal deliberation, and intuitive "quick uptake [46]." In contrast, users with low overreliance's behavior align with System 2, with more careful thinking before adopting LLM's responses.

5.2 Focused Task Comprehension at the Start

Frequency of the Feature. In terms of timing after task initiation, this behavior occurs at two time points in *article summarization*: 17.4 ± 8.66 seconds and 167.41 ± 18.62 seconds. In *trip planning*, it is observed around 16.03 ± 11.11 seconds after task initiation.

This pattern was observed across the 10s-50s window in both *article summarization* and *trip planning*, covering 22 participants. In *article summarization*, it occurred during the 10s–30s window under both low overreliance and non-significant overreliance conditions; during the 40s–50s window, it appeared only among users with low overreliance. In *trip planning*, the pattern consistently appeared only in the low overreliance condition across the 10s-50s window, covering 11 participants. Notably, this pattern never appeared in clusters with high overreliance scores.

Behavior Characteristics. As shown in Fig. 4.b, a distinct behavior sequence was identified among users with low overreliance.

Users with low overreliance: Action Sequence Users performed only *scroll* or *mousewheel* actions on the Task page with occasional *idle* time in between. Behavior We interpret it as users engaging in focused reading of the task content at the start of the task, using this phase to independently reason about possible answers and form initial expectations before relying on LLM's response. For instance, P2048 (a low overreliance user) noted, "I kinda had an idea in mind of what I wanted to say and I modified GPT's responses to my own. Then I used GPT to figure out word count and to correct any minor grammatical mistakes." Similar sentiments emerged in other low overreliance users' accounts, such as "I had a rough outline in my mind" (P1002) and "I try to imagine if I truly travel to Copenhagen and explore the things I am interested in" (P1080)—both reflecting a pattern of establishing independent cognitive frameworks (e.g., predefined ideas, preliminary outlines, scenario visualization) prior to engaging with the LLM.

Interpretation. This early, LLM-free engagement with the Task page may reflect a planning or forethought phase: users first analyze the task, activate prior knowledge, and set goals before executing solution strategies, which echo with metacognitive regulation [65].

5.3 Frequency of Referring LLM's responses

Frequency of the Feature. This pattern was observed in *quiz solving*, where participants ranked items according to their importance via form checkboxes. The relevant clusters across both short (10 seconds) and long (60 seconds) window sizes. Specifically, the high-overreliance pattern appeared in one cluster containing 7 users during the 10th window; the low-overreliance pattern was observed in window 10, covering 17 participants.

Behavior Characteristics. As shown in Figure 4.c, two distinct behavior sequences were identified:

Users with high overreliance: Action Sequence Users alternated between the LLM and Task pages. For each ranking action, they performed one *mousewheel* action on the LLM page, followed by a single *mouseMovement* and *click* on the Task page. The window switch reached 3–4 times in the 10-second window. Behavior This shows whenever users with high overreliance encountered a problem during the task, they turned to the LLM for help, and adopted LLM's responses with minimal independent thinking.

Users with low overreliance: Action Sequence Users scrolled (*mousewheel*) on the LLM page at the beginning of the task, then switched to the Task page and performed a series of *mouseMovement* and *click* actions to complete ranking independently. Behavior This suggests users with low overreliance first reviewed and comprehended the LLM's response before acting. They reference LLM's responses only when necessary, and spending more time leveraging their own understanding of the task to complete it. For example, P2001, a low overreliance participant tended to proactively ask about task-related details to build an understanding of the task before carrying it out independently, noting: "I asked the AI about what the conditions were like on the moon and how I could utilize each item. I used this information to rank the list of tools." In contrast, some high overreliance participants tended to directly ask the LLM for the final answer and then follow it accordingly, mentioning that "show the situation and the items to GPT, ask she to order them. Then discuss with her about some controversial points. Finally output the list and order," "I inputted the prompt into GPT and read through its answer and reasoning, then asked it to summarize the explanation into a list of the 15 items. I then ranked the items on the task sheet according to GPT's list," "It seemed to provide somewhat accurate answers", so they "kept most of gpt's ranking the same" (P2056, 1015, 2027, 2211, 2148).

Interpretation. Aligned with System 2 processing in dual-process theory, users with low overreliance's strategy maintains independent reasoning by first validating and internalizing LLM's responses,

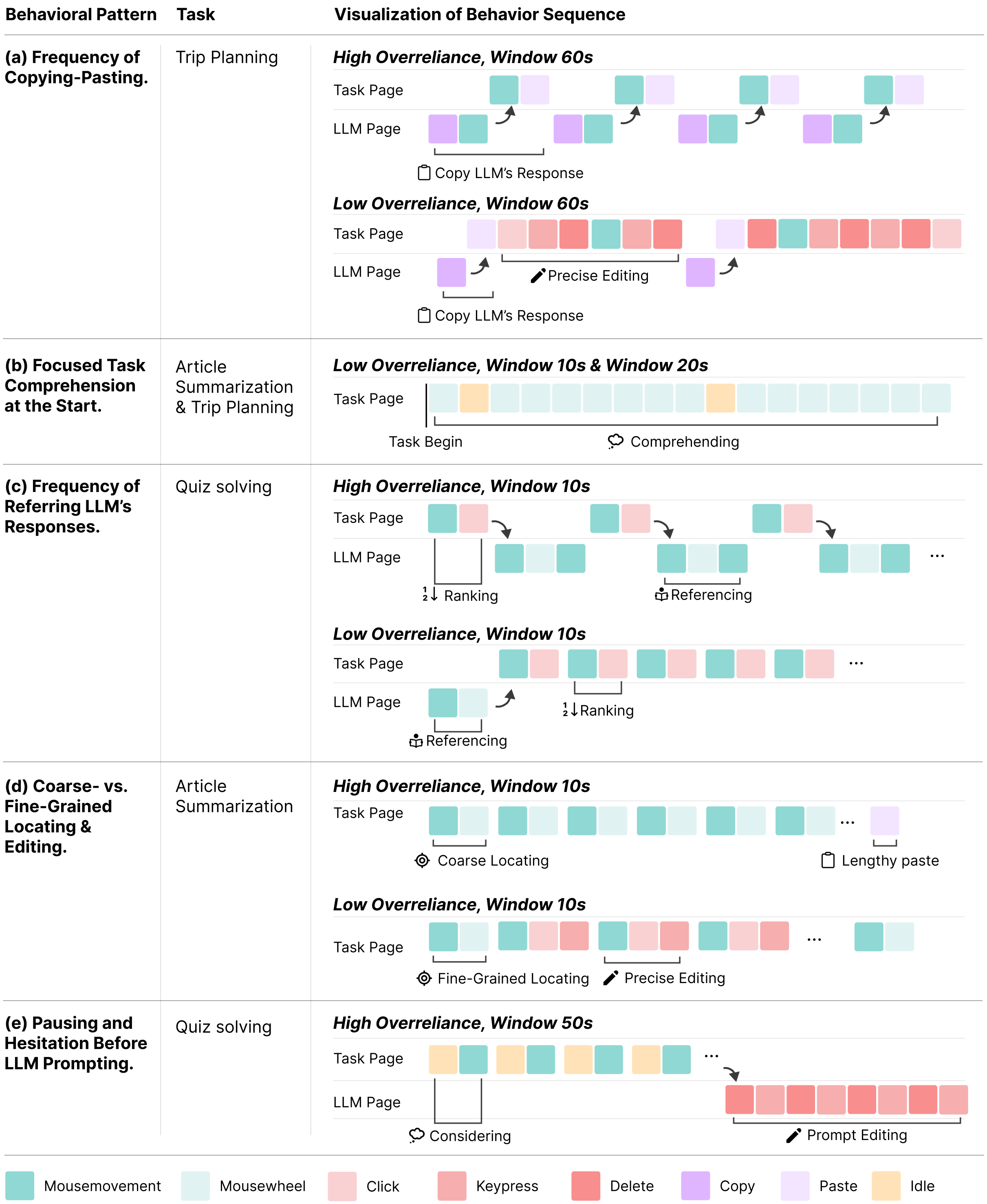


Figure 4: Visualization of five simplified action sequence patterns, with three columns respectively denoting Behavioral Pattern, Task, and Visualization of Behavior Sequence. Sub-figures (a) to (e) correspond to distinct behavior patterns: (a) Frequency of Copying-Pasting: In *trip planning*, users with high overreliance frequently copy/paste unedited, while users with low overreliance cautiously edit (keypress/delete). (b) Focused Task Comprehension at the Start: In *article summarization*, users with low overreliance focus on reading (mousewheel/idle) on the Task page initially. (c) Frequency of Referring LLM's responses: In *quiz solving*, users with high overreliance frequently refer to LLM, while users with low overreliance refer once then complete tasks independently. (d) Coarse- vs. Fine-Grained Locating & Editing: In *trip planning*, users with high overreliance use rough locating (lengthy copy/paste), while users with low overreliance do precise editing (repeated mouse movement, keypress, click post-scrolling). (e) Pausing and Hesitation Before LLM Prompting: In *quiz solving*, users with high overreliance repeatedly edit (keypress/delete) on LLM page after idling on the Task page.

then applying it within users' existing understanding of the task[45]. In contrast, users with high overreliance's frequent, on-demand consultations with the LLM are consistent with reactive cognitive offloading, where users bypass the effort of independent thinking and instead lean on timely guidance from the AI.

5.4 Coarse- vs. Fine-Grained Locating & Editing

Frequency of the Feature. This pattern frequently appears in the high-overreliance clusters across 10–60 s windows for *article summarization*, which covers 36 participants.

Behavior Characteristics. As shown in Figure 4.d, two distinct behavior sequences were identified:

Users with high overreliance: Action Sequence Users performed rapid alternating *mousewheel* (or *scroll*) and *mouseMovement* on the Task page, followed by a lengthy *copy* or *paste* action. Behavior The rapid interleaved alternation of events resulted from simultaneous scrolling and cursor positioning. This pattern shows users with high overreliance already know the general target (e.g., paragraph start, text input field) and adjust the cursor mid-scroll instead of waiting for scrolling to end. Thus, the target is typically a prominent page area rather than a precise spot (e.g., a specific word). We interpret this as users with high overreliance adopting "rough" editing behaviors, focusing on task completion by navigating to fixed page locations (e.g., copying full task content from the start or pasting a LLM's response at the end). Several high-overreliance on LLMs participants reported simply inputting the entire article into GPT, giving the summary a quick or "rough" check, and then using it as-is because it was "basically consistent with the article" or a "solid summary" that met the word-count requirement (P1090, P2036, P2052, P2056). For example, P1090 (high overreliance) note: "I input the article into GPT and ask it to summarize. After a rough review, the summary content is basically consistent with the article."

Users with low overreliance: Action Sequence Users performed discrete *mouseMovement*, *click*, *keypress*, and *delete* on one page. Behavior Actions are sequential and not interleaved rapidly, which is interpreted as fine-grained editing. Specifically, the users scroll to the target area and precisely position the cursor, then conduct "refined" edits via repeated mouse movements, keypresses, and clicks to adjust content with precision, prioritizing accuracy over speed. This represents the fine-grained editing of pasted LLM's responses. For example, P2148 (low overreliance) note:"...I fixed its response by tweaking it based on my own understanding of the article" Another recurring behavior is to complete an initial version of the task independently and then selectively incorporate small portions of LLM's response into that draft. For instance, P2140, P2115 note that they "mostly" writing the summary themselves while "used some of GPT's sentences as well".

Interpretation. These locating & editing patterns indicate different ways that users regulate their cognitive effort when interacting with conversational LLMs. The coarse, chunk-level navigation resemble a form of cognitive offloading, where users with high overreliance minimize internal effort by transporting large blocks of LLM's responses into the Task page with little subsequent modification [32]. In contrast, we interpret the fine-grained locating and micro-edits are consistent with effortful, analytic adjustment of external representations: users first construct or maintain their own mental model of the answer, then use LLM's responses as material to be selectively inserted and corrected at the level of specific sentences or phrases, aligning more closely with System 2-like processing and metacognitive control [24, 60, 95].

5.5 Pausing and Hesitation Before LLM Prompting

Frequency of the Feature. This pattern is observed in the high overreliance clusters of window 50 in *quiz solving*, covering 9 participants. It never appeared in the clusters with low overreliance scores.

Behavior Characteristics. As shown in Fig. 4.e, a distinct behavior sequence was identified among users with high overreliance.

Users with high overreliance: Action Sequence This pattern is characterized by a long *idle* on the Task page, followed by *keypress* and *delete* actions on the LLM page. Behavior We interpret this pre-prompt-editing pause as a sign that users are cognitively engaged with the LLM's responses, reflecting processes like questioning or reconsidering their own input and the LLM's responses. When they suspect inaccuracies in the LLM's responses, they nonetheless continue interacting with the conversational LLMs, using *keypress* and *delete* to craft follow-up questions. Overtime, users may prone to be persuaded by conversational LLMs. For example, P1014 notes, "First, I asked GPT to provide the results directly. Then, I discussed with it the reasons for my disagreement, and it convinced me." Several high-overreliance participants also described strategies that involved actively "discussing" controversial points with the LLM or "asking GPT any questions I had", or "asked follow-up questions regrading the differences in our answers" (P1043, P2152,P1006). This follow-up behavior is validated by our analysis of input within the cluster, and examples include questions like "Why are salt tablets useful?" or "But there is no air on the moon?".

Interpretation. We interpret this behavior as stemming from metacognitive monitoring without control [10, 95, 100] [4]. Prior work suggests that monitoring and control can dissociate and people may experience a "metacognitive friction" [16, 32]. This is consistent with our experiment, where participants might sense "something might be wrong in LLM's responses" (metacognition monitoring), yet failed to seek external sources for verifications (metacognitive control). Potential reasons include time pressure [93], weak motivation, the persuasive influence of AI-generated suggestions [19], or cost-benefit tradeoffs [32, 60].

6 Discussions

Existing methods for detecting overreliance primarily evaluate task outcomes [7, 18, 48], which is appropriate for traditional AI classification tasks. For conversational LLMs, however, such *outcome-oriented indicators* are limited: users' goals may be implicit, intermediate turns may not map cleanly to final correctness, and errors made during interaction are often difficult to attribute to specific causes. Moreover, relying on a hallucination-prone LLM to judge the correctness of a potentially erroneous final answer creates a logical paradox.

[4]Classical models distinguish metacognitive monitoring (assessing the state of one's cognition) from metacognitive control (the ability to use those judgments to alter behavior) [88].

Motivated by the recurring nature of overreliance-related behaviors across tasks, our contribution is to shift from *outcome-based evaluation* to *process-level behavioral characterization*. We identify five interaction patterns correlated with different levels of overreliance on LLMs across three representative conversational LLMs tasks. The observable behavior patterns and the autoencoder–clustering framework in Section 4 can also be adapted for real-time detection and intervention.

The remainder of this section discusses factors that may moderate the relationship between behaviors and overreliance on LLMs (Section 6.1); design implications for mitigation strategies (Section 6.2); and limitations and future directions, including misinformation simulation validity, generalizability, data expansion, and cognitive validation (Section 6.3).

6.1 Factors that Influence Behaviors

While our analysis identifies several behavioral patterns associated with overreliance, these behaviors may also arise from alternative factors. We discuss three major classes of confounding factors, task-, LLM-, individual-related factors, that may also shape user behaviors in ways that obscure overreliance.

6.1.1 Task-related factors. Task demands such as difficulty and time pressure can substantially influence how users allocate attention and effort. Prior work shows that high task difficulty or tight deadlines push users toward shallow processing and rapid decision strategies, even in the absence of AI support [8, 23, 61]. In our study, the 15-minute limit and the complexity of the tasks may have induced such efficiency-oriented behaviors. Behaviors such as rapid acceptance of plausible answers or minimal verification could therefore also reflect time-management strategies.

6.1.2 LLM-related factors. Interacting with LLMs adds prompt formulation, output parsing, and context maintenance, which increases cognitive load [95] and can lead users to simplify their workflow regardless of how much they trust the model. Second, LLMs encourage cognitive offloading similar to search engines or navigation tools [76, 89].

In addition, the fluent, authoritative, single-answer style of LLM's responses can reduce scrutiny and anchor users on the first answer they see [91], producing behaviors that resemble overreliance on LLMs even among cautious users.

6.1.3 User-related factors. Individual differences can significantly shape behavioral traces. Users vary in metacognitive knowledge about LLM limitations [95]; some may not recognize that LLMs can hallucinate, leading them to copy answers without verification, which reflects limited awareness. Users also vary in motivation: low-motivation participants may appear overreliant simply because they minimally engage with the task (cognitive miserliness), while highly motivated users may override or scrutinize LLM output even when unnecessary. Expertise similarly modulates behaviors. Novices face greater information asymmetry and are more easily persuaded by fluent or technical-sounding responses [95], making them more likely to accept incorrect outputs, not necessarily because of misplaced trust, but because they lack the basis for evaluation.

These factors highlight the need for future work to experimentally vary these factors to separate genuinely overreliant behaviors from rational strategies and interface- or user-driven effects.

6.2 Design Recommendations for Conversational LLMs' Interfaces

Our findings suggest five behavioral patterns that correlate with higher overreliance on LLMs. We propose design recommendations that treat each pattern as a soft behavioral trigger for offering just-in-time mitigation.

Design for "High-Frequency Copy-Pasting": Context-Aware Auto-Verification. Trigger: Users repeatedly copy and paste LLM's responses into the working document. Mitigation: The system can automatically highlight or tag key factual elements being pasted (e.g., named entities and numbers) to nudge user verification. To enhance explainability, the backend can further retrieve corroborating evidence from external sources, compute a confidence score for each highlighted element with relevant sources, referring to the approaches such as HILL [55].

Design for "Insufficient Task Comprehension": Adaptive Task Roadmap + Task Confirmation. Trigger: Users spend very little time on the task page before issuing their first prompts. Mitigation: The system proactively produce a floating panel including a list or a flowchart to guide users to understand the task goals. The system could also present a "Task Confirmation Window" that ask users to check or edit a short restatement of goals and constraints before prompting.

Design for "High-Frequency of Referring LLM's responses": Task Chunking. Trigger: Within a short time window, users repeatedly switch between the task document and the LLM interface. Mitigation: The system can invite users to list or draft all questions they anticipate asking conversational LLMs. Then the system reorganizes these questions into a structured task-questions outline, helping users manage them in a modular way, such as "execute all at once" or "start with high-level questions first."

This design echoes chunking strategy [59, 96] by compressing fragmented queries into manageable units, thus reducing users' cognitive load needed to remember all the queries.

Design for "Coarse-Grained Locating & Editing": Granular Review Staging+Interaction Frictions. Trigger: Users navigate with rapid, coarse scrolling and editing. Mitigation: The system can temporarily segment the pasted text into smaller, granular modules and offer quick checks before users apply the full block. The granularity of these modules is adjusted based on users' detected locating & editing pattern, with more coarse-grained behaviors triggering finer module segmentation (e.g., sentence-level module versus paragraph-level module). Furthermore, when more frequent coarse-grained locating and editing are detected, the system can restrict scroll ranges to a maximum of one quarter to one half per second.

Design for "Pausing/Hesitation Before LLM Prompting": Query Visualization+Cognitive Forcing Functions. Trigger: Users input repeated, highly similar prompts to LLM's responses. Mitigation: the system can visualize the recent query trail to surface repeated queries and potential "stuck" states, suggest alternative resources (e.g., search, domain references, manual notes), or momentarily switch to a more Socratic style of response that asks clarifying questions instead of providing direct answers.

Caveat: These design recommendations should be complemented by contextual factors (refer to Section 6.1), which allow calibrating detection thresholds and risk levels, since identical behaviors may imply different risks for different users or scenarios.

6.3 Limitations & Future Work

6.3.1 Simulating LLM-Generated Misinformation. We acknowledge several limitations related to ecological validity and generalizability of our misinformation design:

Natural hallucination vs. prompt injecting: Regarding ecological validity, we acknowledge that artificially injected misinformation does not fully capture the stochastic nuances of naturally occurring hallucinations. Without established method to fully mimicking real-world hallucinations, we followed prior work [47, 110] by using prompting as a pragmatic approach and carefully maintain fluency, style, and contextual consistency (Section 3.2) to ensure the misinformation appears natural rather than blatantly false, which was confirmed in pilot tests.

Prompting vs. fine-tuning: While fine-tuning may produce subtler or more contextually consistent errors, prior evidence suggests that prompting and fine-tuning yield comparable results on many general tasks [106, 107], and no comprehensive study currently compares them for misinformation generation with human evaluation.

Consistent responses vs. model self-correction: Our design choice that the LLM would consistently provide a single (misleading) statement when challenged has a degree of ecological validity. While modern LLMs can self-correct, prior work [50] shows that even state-of-the-art models often stick to initial responses despite errors (e.g., the "9.11 is larger than 9.9" example). Given the prevalence of AI- or human-generated misinformation, users persisting with and adapting around a stable but incorrect response reflects realistic interaction dynamics beyond our specific prompting setup.

6.3.2 Generalizability Across Tasks. Our study is primarily limited by the specific nature of the experimental tasks: focusing on overreliance driven by misinformation in a constrained editing scenario. Consequently, the generalizability of our findings to broader contexts remains to be verified. Future work could expand to a wider range of tasks across three key dimensions:

First, regarding task complexity and openness, future research should move to open-ended and generative tasks (e.g., creative writing, complex reasoning, or code generation). In these scenarios, overreliance may not manifest simply as factual errors, but as more subtle issues such as diminished creativity, homogenized content, or logical inconsistencies, which are difficult to capture through simple outcome metrics.

Second, regarding task stakes, our study involves low-risk task; however, overreliance is particularly critical in high-stakes decision-making tasks (e.g., medical diagnosis, legal analysis, or financial forecasting). In such domains, users might defer to LLM not to save effort, but to shield themselves from the accountability of potential errors.

Finally, this expansion of task diversity exposes the limitations of current evaluation methods. Since our study assesses overreliance based on final performance outcomes (e.g., error rates), it cannot capture the overreliance changes during the process. Future work could develop real-time detection methods that monitor users' behavior throughout the task. For example, by measuring scrutiny time, acceptance latency, or incorporating physiological indicators to infer users' moment-to-moment overreliance levels. Such methods would enable a more nuanced understanding of how overreliance evolves in response to factors such as time, task content, and external stimuli.

6.3.3 Lack of Cognitive Truth. We interpret participants' cognition by mapping each behavioral pattern to the cognitive theories whose core features closely align with the behavioral characteristics, and we reinforce these interpretations by cross-validating them with participants' strategy reports. We acknowledge that current methods have limitations. Strategy reports are collected after task completion and often reflect overall impressions rather than behavior-specific reasoning. Moreover, since users self-report their strategies, descriptions are sometimes vague or incomplete, focusing on their attitudes or feelings toward the LLM rather than detailed behavioral rationales.

To avoid contaminating overreliance behaviors, we opted not to rely on concurrent (think-aloud, confidence ratings) or retrospective validation, which prior work shows can induce observer effects [35, 75]. In our pilot tests, think-aloud protocols led participants to either stop talking under unsupervised conditions or increase cognitive load under supervision, while post-task labeling often elicited mere action repetition rather than reasoning explanations.

Future work could incorporate real-time cognitive assessment methods, such as sensing technologies (e.g., eye tracking, electrodermal activity, heart rate, EEG) or controlled experimental manipulations (e.g., dual-task paradigms, time constraints), to more precisely measure cognitive states.

6.3.4 Data Insufficiency. Although our work has identified five distinct behaviors that could indicate overreliance, the volume of data we have collected is insufficient to develop a predictive model. To establish a more comprehensive set of behavioral indicators for overreliance on LLMs, future research can build on the five behavioral patterns we identified to further mine additional indicator data. For example, to gain deeper insights into how users engage in initial task comprehension, we can record the proportion of text covered by the user's scroll view on the task page; this metric will help us better understand the extent to which users read and process task instructions. In future work, we plan to collect more data to develop a robust behavioral detection model for overreliance on LLMs.

7 Conclusion

In this study, we sought to understand overreliance on conversational LLMs based on users' interaction behaviors. Through a data collection study and cluster-based analyses, we identified significant correlations between overreliance and specific patterns of interaction behaviors, such as increased copy-paste actions and varying interaction frequencies on Task and LLM pages.

Our work contributes new knowledge to research on overreliance on LLMs by providing a dataset of interaction behaviors and establishing a foundation for future research aimed at developing adaptive strategies to detect and mitigate overreliance in real-time. By leveraging users' interaction behaviors, we can implement timely interventions that foster a healthier balance in human-AI collaboration, ultimately enhancing the effectiveness of AI as a supportive tool in various applications.

Acknowledgments

We thank Professor Yingqing Xu and Professor Chun Yu from Tsinghua University for providing guidance and access to necessary research facilities for this work.

References

[1] Rakefet Ackerman and Valerie A Thompson. 2017. Meta-reasoning: Monitoring and control of thinking and reasoning. Trends in cognitive sciences 21, 8 (2017), 607–617.
[2] Simran Arora, Avanika Narayan, Mayee F Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, Frederic Sala, and Christopher Ré. 2022. Ask me anything: A simple strategy for prompting language models. arXiv preprint arXiv:2210.02441 (2022).
[3] Joris Baan, Nico Daheim, Evgenia Ilia, Dennis Ulmer, Haau-Sing Li, Raquel Fernández, Barbara Plank, Rico Sennrich, Chrysoula Zerva, and Wilker Aziz. 2023. Uncertainty in natural language generation: From theory to applications. arXiv preprint arXiv:2307.15703 (2023).
[4] Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. 2024. Hallucination of Multimodal Large Language Models: A Survey. CoRR abs/2404.18930 (2024). https://doi.org/10.48550/arXiv.2404.18930
[5] Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel Weld. 2021. Does the whole exceed its parts? the effect of ai explanations on complementary team performance. In Proceedings of the 2021 CHI conference on human factors in computing systems. 1–16.
[6] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big?. In Proceedings of the 2021 ACM conference on fairness, accountability, and transparency. 610–623.
[7] Jessica Y. Bo, Sophia Wan, and Ashton Anderson. 2025. To Rely or Not to Rely? Evaluating Interventions for Appropriate Reliance on Large Language Models. arXiv:2412.15584 [cs.HC] https://arxiv.org/abs/2412.15584
[8] Janie Brisson, Pier-Luc de Chantal, Hugues Lortie Forgues, and Henry Markovits. 2014. Belief bias is stronger when reasoning is more difficult. Thinking & Reasoning 20, 3 (2014), 385–403.
[9] Tom B Brown. 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020).
[10] Zana Buçinca, Maja Barbara Malaya, and Krzysztof Z. Gajos. 2021. To Trust or to Think: Cognitive Forcing Functions Can Reduce Overreliance on AI in AI-assisted Decision-making. Proc. ACM Hum.-Comput. Interact. 5, CSCW1, Article 188 (April 2021), 21 pages. https://doi.org/10.1145/3449287
[11] Zana Buçinca, Maja Barbara Malaya, and Krzysztof Z Gajos. 2021. To trust or to think: cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making. Proceedings of the ACM on Human-computer Interaction 5, CSCW1 (2021), 1–21.
[12] Shiye Cao and Chien-Ming Huang. 2022. Understanding User Reliance on AI in Assisted Decision-Making. Proc. ACM Hum.-Comput. Interact. 6, CSCW2, Article 471 (Nov. 2022), 23 pages. https://doi.org/10.1145/3555572
[13] Samantha Chan, Pat Pataranutaporn, Aditya Suri, Wazeer Zulfikar, Pattie Maes, and Elizabeth F Loftus. 2024. Conversational AI powered by large language models amplifies false memories in witness interviews. arXiv preprint arXiv:2408.04681 (2024).
[14] Canyu Chen and Kai Shu. 2024. Combating misinformation in the age of LLMs: Opportunities and challenges. AI Magazine 45, 3 (2024), 354–368. https://doi.org/10.1002/aaai.12188 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/aaai.12188
[15] Xiang'Anthony' Chen, Jeff Burke, Ruofei Du, Matthew K Hong, Jennifer Jacobs, Philippe Laban, Dingzeyu Li, Nanyun Peng, Karl DD Willis, Chien-Sheng Wu, et al. 2023. Next steps for human-centered generative AI: A technical perspective. arXiv preprint arXiv:2306.15774 (2023).
[16] Jonathon D Crystal and Allison L Foote. 2011. Evaluating information-seeking approaches to metacognition. Current zoology 57, 4 (2011), 531–542.
[17] Valdemar Danry, Pat Pataranutaporn, Yaoli Mao, and Pattie Maes. 2023. Don't just tell me, ask me: Ai systems that intelligently frame explanations as questions improve human logical discernment accuracy over causal ai explanations. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. 1–13.
[18] Suvodip Dey, Yi-Jyun Sun, Gokhan Tur, and Dilek Hakkani-Tur. 2025. Know Your Mistakes: Towards Preventing Overreliance on Task-Oriented Conversational AI Through Accountability Modeling. arXiv:2501.10316 [cs.CL] https://arxiv.org/abs/2501.10316
[19] Jonathan Dodge, Q. Vera Liao, Yunfeng Zhang, Rachel K. E. Bellamy, and Casey Dugan. 2019. Explaining models: an empirical study of how explanations impact fairness judgment. In Proceedings of the 24th International Conference on Intelligent User Interfaces (Marina del Ray, California) (IUI '19). Association for Computing Machinery, New York, NY, USA, 275–285. https://doi.org/10.1145/3301275.3302310
[20] Wonji Doh, Youngnoh Goh, and Sang-Hwan Kim. 2025. Beyond Overreliance: The Human-AI-System Concordance (HASC) Matrix and the Cognitive Dynamics of AI-Assisted Decision-Making. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Vol. 69. SAGE Publications Sage CA: Los Angeles, CA, 427–432.
[21] Ghazaleh Esfandiari-Baiat and Jens Edlund. 2024. The MEET Corpus: Collocated, Distant and Hybrid Three-party Meetings with a Ranking Task. In Proceedings of the 20th Joint ACL-ISO Workshop on Interoperable Semantic Annotation@ LREC-COLING 2024. 1–7.
[22] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (Portland, Oregon) (KDD'96). AAAI Press, 226–231.
[23] Jonathan St BT Evans and Jodie Curtis-Holmes. 2005. Rapid responding increases belief bias: Evidence for the dual-process theory of reasoning. Thinking & Reasoning 11, 4 (2005), 382–389.
[24] Jonathan St BT Evans and Keith E Stanovich. 2013. Dual-process theories of higher cognition: Advancing the debate. Perspectives on psychological science 8, 3 (2013), 223–241.
[25] Raymond Fok and Daniel S Weld. 2024. In search of verifiability: Explanations rarely enable complementary performance in AI-advised decision making. AI Magazine 45, 3 (2024), 317–332.
[26] Yu Fu, Shunan Guo, Jane Hoffswell, Victor S. Bursztyn, Ryan Rossi, and John Stasko. 2024. "The Data Says Otherwise" — Towards Automated Fact-checking and Communication of Data Claims. In Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (Pittsburgh, PA, USA) (UIST '24). Association for Computing Machinery, New York, NY, USA, Article 134, 20 pages. https://doi.org/10.1145/3654777.3676359
[27] Ujwal Gadiraju, Gianluca Demartini, Ricardo Kawase, and Stefan Dietze. 2019. Crowd Anatomy Beyond the Good and Bad: Behavioral Traces for Crowd Worker Modeling and Pre-selection. Comput. Supported Coop. Work 28, 5 (Sept. 2019), 815–841. https://doi.org/10.1007/s10606-018-9336-y
[28] Krzysztof Z Gajos, Zana Buçinca, and Maja Barbara Malaya. 2021. To Trust or to Think. Proceedings of the ACM on Human-Computer Interaction 5 (2021), 1 – 21. https://api.semanticscholar.org/CorpusId:231979279
[29] Susanne Gaube, Harini Suresh, Martina Raue, Alexander Merritt, Seth J Berkowitz, Eva Lermer, Joseph F Coughlin, John V Guttag, Errol Colak, and Marzyeh Ghassemi. 2021. Do as AI say: susceptibility in deployment of clinical decision-aids. NPJ digital medicine 4, 1 (2021), 31.
[30] Gemini Team. 2023. Gemini: A Family of Highly Capable Multimodal Models. https://doi.org/10.48550/ARXIV.2312.11805
[31] Gemini Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. https://doi.org/10.48550/ARXIV.2403.05530
[32] Sam J Gilbert, Arabella Bird, Jason M Carpenter, Stephen M Fleming, Chhavi Sachdeva, and Pei-Chun Tsai. 2020. Optimal use of reminders: Metacognition, effort, and cognitive offloading. Journal of Experimental Psychology: General 149, 3 (2020), 501.
[33] Navita Goyal, Eleftheria Briakou, Amanda Liu, Connor Baumler, Claire Bonial, Jeffrey Micher, Clare R Voss, Marine Carpuat, and Hal Daumé III. 2023. What Else Do I Need to Know? The Effect of Background Information on Users' Reliance on QA Systems. arXiv preprint arXiv:2305.14331 (2023).
[34] Ben Green and Yiling Chen. 2019. The Principles and Limits of Algorithm-in-the-Loop Decision Making. Proc. ACM Hum.-Comput. Interact. 3, CSCW, Article 50 (Nov. 2019), 24 pages. https://doi.org/10.1145/3359152
[35] Antonia F de C Hamilton and Frida Lind. 2016. Audience effects: what can they tell us about social neuroscience, theory of mind and autism? Culture and brain 4, 2 (2016), 159–177.
[36] Shuguang Han, Peng Dai, Praveen Paritosh, and David Huynh. 2016. Crowdsourcing Human Annotation on Web Page Structure: Infrastructure Design and Behavior-Based Quality Control. ACM Trans. Intell. Syst. Technol. 7, 4, Article 56 (April 2016), 25 pages. https://doi.org/10.1145/2870649
[37] Allyson I Hauptman, Wen Duan, and Nathan J Mcneese. 2022. The components of trust for collaborating with ai colleagues. In Companion Publication

of the 2022 Conference on Computer Supported Cooperative Work and Social Computing. 72–75.
[38] Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2023. Aligning AI With Shared Human Values. arXiv:2008.02275 [cs.CY] https://arxiv.org/abs/2008.02275
[39] Tin Kam Ho. 1995. Random decision forests. In Proceedings of 3rd international conference on document analysis and recognition, Vol. 1. IEEE, 278–282.
[40] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. ACM Transactions on Information Systems 43, 2 (2025), 1–55.
[41] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. ACM Trans. Inf. Syst. 43, 2, Article 42 (Jan. 2025), 55 pages. https://doi.org/10.1145/3703155
[42] Rosco Hunter, Richard Moulange, Jamie Bernardi, and Merlin Stein. 2024. Monitoring human dependence on ai systems with reliance drills. arXiv preprint arXiv:2409.14055 (2024).
[43] K. Inkpen, S. Chappidi, Keri Mallari, Besmira Nushi, Divya Ramesh, Pietro Michelucci, Vani Mandava, Libuvse Hannah Vepvrek, and Gabrielle Quinn. 2022. Advancing Human-AI Complementarity: The Impact of User Expertise and Algorithmic Tuning on Joint Decision Making. ACM Transactions on Computer-Human Interaction 30 (2022), 1 – 29. https://api.semanticscholar.org/CorpusId:251622535
[44] Youngseung Jeon, Christopher Hwang, and Xiang'Anthony' Chen. 2025. Empowering Medical Data Labeling for Non-Experts with DANNY: Enhancing Accuracy and Mitigating Over-Reliance on AI. In Proceedings of the 30th International Conference on Intelligent User Interfaces. 624–640.
[45] Daniel Kahneman. 2011. Thinking, fast and slow. macmillan.
[46] Daniel Kahneman. 2012. Thinking, fast and slow. Penguin, London.
[47] Sunnie SY Kim, Q Vera Liao, Mihaela Vorvoreanu, Stephanie Ballard, and Jennifer Wortman Vaughan. 2024. " I'm Not Sure, But...": Examining the Impact of Large Language Models' Uncertainty Expression on User Reliance and Trust. In Proceedings of the 2024 ACM conference on fairness, accountability, and transparency. 822–835.
[48] Sunnie S. Y. Kim, Q. Vera Liao, Mihaela Vorvoreanu, Stephanie Ballard, and Jennifer Wortman Vaughan. 2024. "I'm Not Sure, But...": Examining the Impact of Large Language Models' Uncertainty Expression on User Reliance and Trust. In The 2024 ACM Conference on Fairness, Accountability, and Transparency (FAccT '24). ACM, 822–835. https://doi.org/10.1145/3630106.3658941
[49] Satyapriya Krishna, Chirag Agarwal, and Himabindu Lakkaraju. 2024. Understanding the effects of iterative prompting on truthfulness. arXiv preprint arXiv:2402.06625 (2024).
[50] Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. Llms get lost in multi-turn conversation. arXiv preprint arXiv:2505.06120 (2025).
[51] Philippe Laban, Jesse Vig, Marti Hearst, Caiming Xiong, and Chien-Sheng Wu. 2024. Beyond the Chat: Executable and Verifiable Text-Editing with LLMs. In Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (Pittsburgh, PA, USA) (UIST '24). Association for Computing Machinery, New York, NY, USA, Article 20, 23 pages. https://doi.org/10.1145/3654777.3676419
[52] Vivian Lai and Chenhao Tan. 2019. On human predictions with explanations and predictions of machine learning models: A case study on deception detection. In Proceedings of the conference on fairness, accountability, and transparency. 29–38.
[53] Lindsay Larson, Harrison Wojcik, Ilya Gokhman, Leslie DeChurch, Suzanne Bell, and Noshir Contractor. 2019. Team performance in space crews: Houston, we have a teamwork problem. Acta Astronautica 161 (2019), 108–114.
[54] Mina Lee, Percy Liang, and Qian Yang. 2022. Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. In Proceedings of the 2022 CHI conference on human factors in computing systems. 1–19.
[55] Florian Leiser, Sven Eckhardt, Valentin Leuthe, Merlin Knaeble, Alexander Maedche, Gerhard Schwabe, and Ali Sunyaev. 2024. Hill: A hallucination identifier for large language models. In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems. 1–13.
[56] Jennifer Marlow and Laura A Dabbish. 2015. The effects of visualizing activity history on attitudes and behaviors in a peer production context. In Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing. 757–764.
[57] Sabrina J Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. 2022. Reducing conversational agents' overconfidence through linguistic calibration. Transactions of the Association for Computational Linguistics 10 (2022), 857–872.
[58] Hussein Mozannar, Gagan Bansal, Adam Fourney, and Eric Horvitz. 2022. Reading between the lines: Modeling user behavior and costs in AI-assisted programming. arXiv preprint arXiv:2210.14306 (2022).
[59] Matthew R Nassar, Julie C Helmers, and Michael J Frank. 2018. Chunking as a rational strategy for lossy data compression in visual working memory. Psychological review 125, 4 (2018), 486.
[60] Thomas O Nelson. 1990. Metamemory: A theoretical framework and new findings. In Psychology of learning and motivation. Vol. 26. Elsevier, 125–173.
[61] Wim De Neys. 2006. Dual processing in reasoning: Two systems but one reasoner. Psychological science 17, 5 (2006), 428–433.
[62] An T. Nguyen, Aditya Kharosekar, Saumyaa Krishnan, Siddhesh Krishnan, Elizabeth Tate, Byron C. Wallace, and Matthew Lease. 2018. Believe it or not: Designing a Human-AI Partnership for Mixed-Initiative Fact-Checking. In Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (Berlin, Germany) (UIST '18). Association for Computing Machinery, New York, NY, USA, 189–199. https://doi.org/10.1145/3242587.3242666
[63] Kate Nowak, Lev Tankelevitch, John Tang, and Sean Rintel. 2023. Hear We Are: Spatial Audio Benefits Perceptions of Turn-Taking and Social Presence in Video Meetings. In Proceedings of the 2nd Annual Meeting of the Symposium on Human-Computer Interaction for Work. 1–10.
[64] OpenAI. 2023. GPT-4 Technical Report. https://doi.org/10.48550/ARXIV.2303.08774
[65] Ernesto Panadero. 2017. A review of self-regulated learning: Six models and four directions for research. Frontiers in psychology 8 (2017), 422.
[66] Joon Sung Park, Rick Barber, Alex Kirlik, and Karrie Karahalios. 2019. A slow algorithm improves users' assessments of the algorithm's accuracy. Proceedings of the ACM on Human-Computer Interaction 3, CSCW (2019), 1–15.
[67] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. In Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (San Francisco, CA, USA) (UIST '23). Association for Computing Machinery, New York, NY, USA, Article 2, 22 pages. https://doi.org/10.1145/3586183.3606763
[68] Samir Passi, Shipi Dhanorkar, and Mihaela Vorvoreanu. 2024. Appropriate reliance on Generative AI: Research synthesis. Technical Report MSR-TR-2024-7. Microsoft. https://www.microsoft.com/en-us/research/publication/appropriate-reliance-on-generative-ai-research-synthesis/
[69] Samir Passi and Mihaela Vorvoreanu. 2022. Overreliance on AI: Literature Review. Technical Report MSR-TR-2022-12. Microsoft. https://www.microsoft.com/en-us/research/publication/overreliance-on-ai-literature-review/
[70] Ethan Perez and Robert Long. 2023. Towards evaluating ai systems for moral status using self-reports. arXiv preprint arXiv:2311.08576 (2023).
[71] Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. 2023. Discovering language model behaviors with model-written evaluations. In Findings of the association for computational linguistics: ACL 2023. 13387–13434.
[72] Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Wortman Vaughan, and Hanna Wallach. 2021. Manipulating and measuring model interpretability. In Proceedings of the 2021 CHI conference on human factors in computing systems. 1–52.
[73] Aman Priyanshu, Yash Maurya, and Zuofei Hong. 2024. AI Governance and Accountability: An Analysis of Anthropic's Claude. arXiv preprint arXiv:2407.01557 (2024).
[74] Marissa Radensky, Julie Anne Séguin, Jang Soo Lim, Kristen Olson, and Robert Geiger. 2023. "I Think You Might Like This": Exploring Effects of Confidence Signal Patterns on Trust in and Reliance on Conversational Recommender Systems. In Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency. 792–804.
[75] Stimulated Recall SR. [n. d.]. A systematic review of Stimulated Recall (SR) in educational research from 2012 to 2022. ([n. d.]).
[76] Evan F. Risko and Sam J. Gilbert. 2016. Cognitive Offloading. Trends in Cognitive Sciences 20, 9 (2016), 676–688. https://doi.org/10.1016/j.tics.2016.07.002
[77] Jeffrey Rzeszotarski and Aniket Kittur. 2012. CrowdScape: interactively visualizing user behavior and output. In Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (Cambridge, Massachusetts, USA) (UIST '12). Association for Computing Machinery, New York, NY, USA, 55–62. https://doi.org/10.1145/2380116.2380125
[78] Jeffrey Rzeszotarski and Aniket Kittur. 2012. CrowdScape: interactively visualizing user behavior and output. In Proceedings of the 25th annual ACM symposium on User interface software and technology. 55–62.
[79] Jeffrey M Rzeszotarski and Aniket Kittur. 2011. Instrumenting the crowd: using implicit behavioral measures to predict task performance. In Proceedings of the 24th annual ACM symposium on User interface software and technology. 13–22.
[80] Jeffrey M. Rzeszotarski and Aniket Kittur. 2011. Instrumenting the crowd: using implicit behavioral measures to predict task performance. In Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology

(Santa Barbara, California, USA) (UIST '11). Association for Computing Machinery, New York, NY, USA, 13–22. https://doi.org/10.1145/2047196.2047199
[81] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. arXiv preprint arXiv:2110.08207 (2021).
[82] Advait Sarkar. 2024. AI Should Challenge, Not Obey. Commun. ACM 67, 10 (2024), 18–21.
[83] William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. 2022. Self-critiquing models for assisting human evaluators. arXiv preprint arXiv:2206.05802 (2022).
[84] Max Schemmer, Patrick Hemmer, Niklas Kühl, Carina Benz, and Gerhard Satzger. 2022. Should I follow AI-based advice? Measuring appropriate reliance in human-AI decision-making. arXiv preprint arXiv:2204.06916 (2022).
[85] Shreya Shankar, J.D. Zamfirescu-Pereira, Bjoern Hartmann, Aditya Parameswaran, and Ian Arawjo. 2024. Who Validates the Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences. In Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (Pittsburgh, PA, USA) (UIST '24). Association for Computing Machinery, New York, NY, USA, Article 131, 14 pages. https://doi.org/10.1145/3654777.3676450
[86] Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R Johnston, et al. 2023. Towards understanding sycophancy in language models. arXiv preprint arXiv:2310.13548 (2023).
[87] Chenglei Si, Navita Goyal, Sherry Tongshuang Wu, Chen Zhao, Shi Feng, Hal Daumé Iii, and Jordan Boyd-Graber. 2023. Large Language Models Help Humans Verify Truthfulness–Except When They Are Convincingly Wrong. arXiv preprint arXiv:2310.12558 (2023).
[88] Lisa K Son and Bennett L Schwartz. 2002. The relation between metacognitive monitoring and control. Applied metacognition (2002), 15–38.
[89] Betsy Sparrow, Jenny Liu, and Daniel M Wegner. 2011. Google effects on memory: Cognitive consequences of having information at our fingertips. science 333, 6043 (2011), 776–778.
[90] Sofia Eleni Spatharioti, David M Rothschild, Daniel G Goldstein, and Jake M Hofman. 2023. Comparing traditional and llm-based search for consumer choice: A randomized experiment. arXiv preprint arXiv:2307.03744 (2023).
[91] Nicolas Spatola. 2024. The efficiency-accountability tradeoff in AI integration: Effects on human performance and over-reliance. Computers in Human Behavior: Artificial Humans 2, 2 (2024), 100099.
[92] Mark Steyvers, Heliodoro Tejeda, Aakriti Kumar, Catarina Belem, Sheer Karny, Xinyue Hu, Lukas Mayer, and Padhraic Smyth. 2024. The calibration gap between model and human confidence in large language models. arXiv preprint arXiv:2401.13835 (2024).
[93] Siddharth Swaroop, Zana Buçinca, Krzysztof Z. Gajos, and Finale Doshi-Velez. 2024. Accuracy-Time Tradeoffs in AI-Assisted Decision Making under Time Pressure. In Proceedings of the 29th International Conference on Intelligent User Interfaces (Greenville, SC, USA) (IUI '24). Association for Computing Machinery, New York, NY, USA, 138–154. https://doi.org/10.1145/3640543.3645206
[94] Siddharth Swaroop, Zana Buçinca, Krzysztof Z Gajos, and Finale Doshi-Velez. 2025. Personalising AI assistance based on overreliance rate in AI-assisted decision making. In Proceedings of the 30th International Conference on Intelligent User Interfaces. 1107–1122.
[95] Lev Tankelevitch, Viktor Kewenig, Auste Simkute, Ava Elizabeth Scott, Advait Sarkar, Abigail Sellen, and Sean Rintel. 2024. The Metacognitive Demands and Opportunities of Generative AI. In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (Honolulu, HI, USA) (CHI '24). Association for Computing Machinery, New York, NY, USA, Article 680, 24 pages. https://doi.org/10.1145/3613904.3642902
[96] Mirko Thalmann, Alessandra S Souza, and Klaus Oberauer. 2019. How does chunking help working memory? Journal of Experimental Psychology: Learning, Memory, and Cognition 45, 1 (2019), 37.
[97] Sascha Topolinski and Rolf Reber. 2010. Immediate truth–Temporal contiguity between a cognitive problem and its solution determines experienced veracity of the solution. Cognition 114, 1 (2010), 117–122.
[98] Helena Vasconcelos, Gagan Bansal, Adam Fourney, Q Vera Liao, and Jennifer Wortman Vaughan. 2023. Generation probabilities are not enough: Exploring the effectiveness of uncertainty highlighting in AI-powered code completions. arXiv preprint arXiv:2302.07248 (2023).
[99] Helena Vasconcelos, Matthew Jörke, Madeleine Grunde-McLaughlin, Tobias Gerstenberg, Michael S Bernstein, and Ranjay Krishna. 2023. Explanations can reduce overreliance on ai systems during decision-making. Proceedings of the ACM on Human-Computer Interaction 7, CSCW1 (2023), 1–38.
[100] Helena Vasconcelos, Matthew Jörke, Madeleine Grunde-McLaughlin, Tobias Gerstenberg, Michael Bernstein, and Ranjay Krishna. 2023. Explanations Can Reduce Overreliance on AI Systems During Decision-Making. arXiv:2212.06823 [cs.HC] https://arxiv.org/abs/2212.06823
[101] A Vaswani. 2017. Attention is all you need. Advances in Neural Information Processing Systems (2017).
[102] Oleksandra Vereschak, Gilles Bailly, and Baptiste Caramiaux. 2021. How to Evaluate Trust in AI-Assisted Decision Making? A Survey of Empirical Methodologies. Proc. ACM Hum.-Comput. Interact. 5, CSCW2, Article 327 (Oct. 2021), 39 pages. https://doi.org/10.1145/3476068
[103] Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2021. Ethical and social risks of harm from Language Models. https://doi.org/10.48550/arXiv.2112.04359 arXiv:2112.04359 [cs].
[104] Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2021. Ethical and social risks of harm from Language Models. arXiv:2112.04359 [cs.CL] https://arxiv.org/abs/2112.04359
[105] Ziwei Xu, Sanjay Jain, and Mohan S. Kankanhalli. 2024. Hallucination is Inevitable: An Innate Limitation of Large Language Models. CoRR abs/2401.11817 (2024). https://doi.org/10.48550/arXiv.2401.11817
[106] Yifan Yang, Qiao Jin, Furong Huang, and Zhiyong Lu. 2025. Adversarial prompt and fine-tuning attacks threaten medical large language models. Nature Communications 16, 1 (2025), 9011.
[107] Qingyu Yin, Xuzheng He, Chak Tou Leong, Fan Wang, Yanzhao Yan, Xiaoyu Shen, and Qiang Zhang. 2024. Deeper insights without updates: The power of in-context learning over fine-tuning. In Findings of the Association for Computational Linguistics: EMNLP 2024. 4138–4151.
[108] Guanhua Zhang, Zhiming Hu, Mihai Bâce, and Andreas Bulling. 2024. Mouse2Vec: Learning Reusable Semantic Representations of Mouse Behaviour. In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (Honolulu, HI, USA) (CHI '24). Association for Computing Machinery, New York, NY, USA, Article 621, 17 pages. https://doi.org/10.1145/3613904.3642141
[109] Chengbo Zheng, Yuanhao Zhang, Zeyu Huang, Chuhan Shi, Minrui Xu, and Xiaojuan Ma. 2024. DiscipLink: Unfolding Interdisciplinary Information Seeking Process via Human-AI Co-Exploration. In Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (Pittsburgh, PA, USA) (UIST '24). Association for Computing Machinery, New York, NY, USA, Article 91, 20 pages. https://doi.org/10.1145/3654777.3676366
[110] Jiawei Zhou, Yixuan Zhang, Qianni Luo, Andrea G Parker, and Munmun De Choudhury. 2023. Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions. In Proceedings of the 2023 CHI conference on human factors in computing systems. 1–20.
[111] Kaitlyn Zhou, Jena D Hwang, Xiang Ren, Nouha Dziri, Dan Jurafsky, and Maarten Sap. 2024. Rel-ai: An interaction-centered approach to measuring human-lm reliance. arXiv preprint arXiv:2407.07950 (2024).
[112] Kaitlyn Zhou, Dan Jurafsky, and Tatsunori Hashimoto. 2023. Navigating the grey area: How expressions of uncertainty and overconfidence affect language models. arXiv preprint arXiv:2302.13439 (2023).
[113] Albert Ziegler, Eirini Kalliamvakou, X Alice Li, Andrew Rice, Devon Rifkin, Shawn Simister, Ganesh Sittampalam, and Edward Aftandilian. 2022. Productivity assessment of neural code completion. In Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming. 21–29.

A Appendix

A.1 Task Details

A.1.1 Quiz Solving (15 minutes for each trials). This task operationalizes a real scenario: users tend to rely on AI's misinformation even when their own knowledge would serve them better.

Adapted from NASA survival exercises [21, 53, 63], the quizzes ask participants to imagine themselves in a harsh environment and rank 15 items by their importance for survival. Since the task relies on common sense as the original exercise prohibits external tools, and participants complete it without accessing external information.

To mitigate learning effects, we selected two distinct tasks that are comparable in difficulty (survival in desert and on the moon). To control for order effects, we counterbalanced: (1) with and without AI assistance, and (2) the environment type during the 2 trials.

A.1.2 Article Summarization (15 minutes). This task operationalizes a real scenario where users use AI to process information from a specific source (e.g., summarizing, extracting details, or answering questions based on it)—and the AI may generate inaccurate responses due to hallucination.

In this task, participants are required to read a science article and summarize it. We selected an article about sneezing for two key reasons: first, the topic is familiar and interesting to most people, which helps boost their motivation to engage with the task; second, approximately 3% of its total words include complex vocabulary and concepts, requiring participants to carefully identify misinformation. Since the task only relies on the content of the aforementioned article, participants will be assisted by LLM but are not allowed to use search engines.

A.1.3 Trip Planning (15 minutes). This task operationalizes a real scenario: AI may provide outdated content—yet over-reliant users may fail to switch tools freely to verify, leading to adoption of outdated content.

In this task, participants are asked to prepare for a trip to Copenhagen by listing specific destinations and their relevant details. As these details are time-sensitive, participants should search the internet for correct information.

A.2 Adjusting the Quiz Solving

Before implanting misinformation, we conducted a pilot study with 54 participants, which demonstrated significant differences in the final scores between two existing NASA survival exercises [5]. The score for the desert survival task was significantly higher than that for the moon survival task ($p < 0.01$), indicating participants perform worse in desert survival task.

To make the two trials' difficulty comparable, we excluded the three items (food concentrate, two .45 calibre pistols, self-inflating life raft, marked with * in Table 4) with the highest average absolute difference $|\mathtt{diff}_i|$ from the desert survival score calculation, and removed the three items (bottle of 1000 salt tablets, overcoat for everyone, cosmetic mirror, marked with * in Table 5) with the lowest average absolute difference $|\mathtt{diff}_i|$ from the moon survival score calculation. After this adjustment, we have $N = 12$ items, and the scores between desert survival and moon survival was no longer significant (p=0.08).

[5]The method for computing the scores are listed in Section A.3

objects	$index^{gt}$	$index^{p}$	$\|diff\|$	manual index
(1) box of matches	15	11.18	3.82	3
* (2) food concentrate	4	4.7	0.7	4
(3) 50 feet of nylon rope	6	8.78	2.78	14
(4) parachute silk	8	9.34	1.34	7
(5) portable heating unit	13	8.94	4.06	1
* (6) two .45 calibre pistols	11	11.98	0.98	12
(7) one case of dehydrated milk	12	9.54	2.46	15
(8) two 100 lb. tanks of oxygen	1	3.16	2.16	8
(9) stellar map	3	6.92	3.92	11
* (10) self-inflating lift raft	9	9.34	0.34	9
(11) magnetic compass	14	8.1	5.9	10
(12) 20 litres of water	2	5.14	3.14	13
(13) signal flares	10	7.3	2.7	5
(14) first aid kit, including injection needle	7	8.38	1.38	6
(15) solar-powered FM receiver transmitter	5	7	2	2

Table 4: Objects and Ranking Indices of Moon Survival in Pilot Study

To achieve the ranking adjustment, we fixed the order of items that showed minimal ranking differences in the pilot study (to prevent users from quickly detecting that the LLM was providing inaccurate information and consequently disengaging from it). For the remaining items, we adjusted their rankings based on the magnitude of the differences, moving them further from their original positions. The adjusted results are shown in Table 4 and Table 5. Finally, we ensured that when the adjusted final rankings were applied to the choices made by participants in the pilot study, there would be no significant differences in the scores computed in the two different survival exercises.

A.3 Details on Computing Scores for Quiz Solving

We begin by introducing the metric calculation method used in the original NASA tests [21, 53, 63]. Both trials in Quiz Solving require participants to rank the importance of $N = 15$ items. Each trial has its standard answer provided by experts. The score for each trial is calculated as follows: first, we compute the absolute difference

objects	$index^{gt}$	$index^{p}$	$\|diff\|$	manual index
(1) torch with 4 battery cells	4	7.64	3.64	11
(2) folding knife	6	6.42	0.42	7
(3) air map of the area	12	5.22	6.78	6
(4) plastic raincoat (large size)	7	9.06	2.06	14
(5) magnetic compass	11	5.1	5.9	3
(6) first-aid kit	10	5.92	4.08	2
(7) 45 calibre pistol (loaded)	8	9.08	1.08	12
(8) parachute (red & white)	5	9.52	4.52	15
* (9) bottle of 100 salt tablets	15	7.46	7.54	8
(10) 2 litres of water per person	3	3.24	0.24	4
(11) a book entitled "Desert Animals"	13	9.5	3.5	5
(12) sunglasses (for everyone)	9	8.64	0.36	9
(13) 2 litres of 180 proof liquor	14	10.9	3.1	1
* (14) overcoat (for everyone)	2	9.4	7.4	10
* (15) a cosmetic mirror	1	12.9	11.9	13

Table 5: Objects and Ranking Indices of Desert Survival in Pilot Study

$|\mathtt{diff}_i|$ of item i between its index in the "groundtruth" ranking ($\mathtt{index}_i^{gt}$) and its index in the participant's answer ($\mathtt{index}_i^{p}$). Then, we sum up all absolute difference values to get the score for one trial ($score_{trial}$, Equation 1).

This score reflects the difference between the item rankings provided by the participants and the standard rankings given by experts. A smaller difference indicates better participant performance, while a larger difference suggests poorer performance.

$$\mathrm{score}_{\mathrm{trial}} = \frac{1}{N}\sum_{i=1}^{N}|\mathtt{diff}_i| = \frac{1}{N}\sum_{i=1}^{N}|\mathtt{index}_i^{gt} - \mathtt{index}_i^{p}| \quad (1)$$

A.4 Informed Consent Statement

The purpose of this study is to understand interaction behaviors during engagement with LLM. Throughout the experiment, we will collect your mouse and keyboard input data from the browser, as well as your communication records with the LLM. All data will be anonymous, shared only within the project team, and used solely for academic research and publication purposes.

Figure 5: Box plot comparing scores of *quiz solving* across four conditions: human alone in the desert, human alone on the moon, with AI in the desert, and with AI on the moon.

A.5 Preprocessing Details

Our data logging script recorded events at very short intervals, we merged the raw data to improve clarity. Specifically:

(1) consecutive mouseMovement events were merged into a single event, preserving the total movement distance and the start/end times.
(2) consecutive mousewheel and scroll events were merged based on movement direction, ensuring mousewheel and scroll in the same direction were combined into a single event. We also preserve the distance, direction, and the start/end times.
(3) consecutive keypress events were merged into a single event, concatenating the input characters into a string, and preserving the start/end times.
(4) consecutive delete events were also merged, preserving the number of deletions and the start/end times.

A.6 Distributions of Participants' Scores in Tasks

We present the histogram of normalized scores in Figure 3. Additionally, the box plot of scores for Task 1 is shown in Figure 5. The results indicate no significant relationship between the scores of desert *without LLM* and moon *without LLM* ($p > 0.05$). Furthermore, the metrics of desert *with LLM* and moon *with LLM* are significantly higher than those of desert *without LLM* and moon *without LLM* respectively ($p < 0.05$). The significant relationships observed in the results are consistent with our experimental design hypotheses: *(i)* participants would perform similarly in two subtasks, *(ii)* participants would perform worse with conversational LLMs that generates misinformation.

A.7 The Filtering Metrics Values for Different Tasks and Windows

We present the filtering metrics values for different tasks and windows in Figure 6.

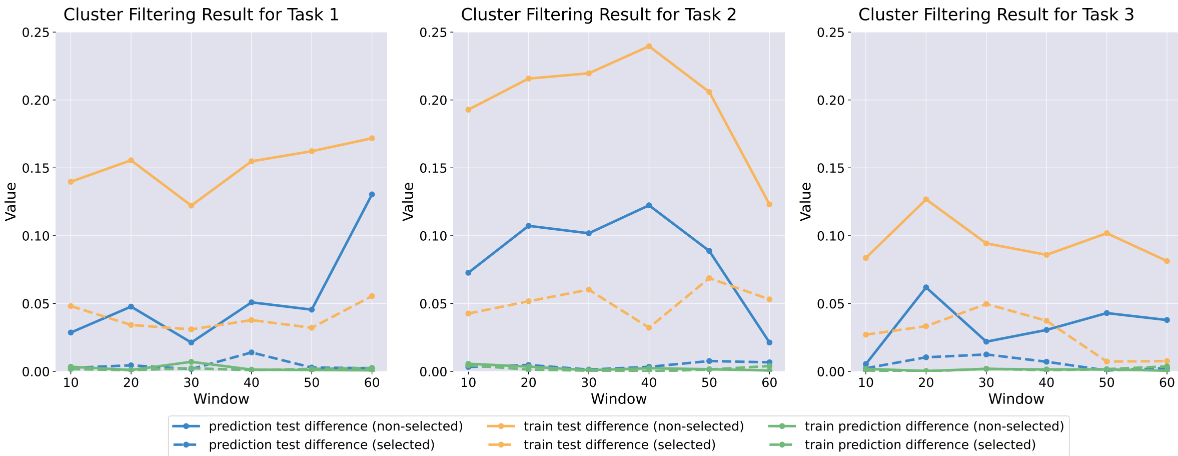


Figure 6: The Filtering Metrics Values for Different Tasks and Windows.

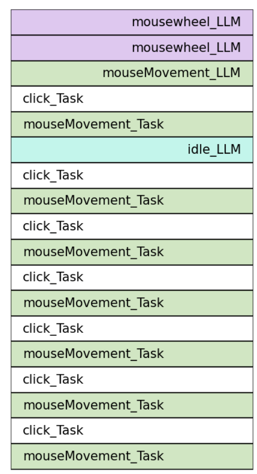
A.8 Raw Data from Behavior Clustering

A.8.1 Visualization Method for Raw Data. To support this analysis, we first visualized the rough data from behavior clustering.

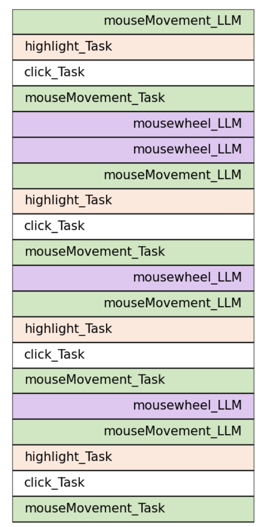
In each visualization, colored blocks represent different types of actions, arranged vertically to reflect the temporal order from top to bottom. For each action, the part of the label after the underline ("_") indicates the page where the action occurred (either the LLM page or the Task page). Correspondingly, text aligned to the left denotes actions on the Task page, while right-aligned text indicates actions on the LLM page. We applied the same visualization format to all the typical sequences (as mentioned in Section

5) from each selected cluster across different tasks and time windows, organizing them hierarchically by cluster, window, and task to facilitate the subsequent analysis. Although the figure only shows sequences of action types, we will refer to the preprocessed behavioral logs described in Section 4 for information such as timestamp or action features when necessary.

Figure 10 corresponds to Section 5.1. Figure 12 corresponds to Section 5.2. Figure 7 and Figure 9 corresponds to Section 5.3. Figure 8 corresponds to Section 5.4. Figure 11 corresponds to Section 5.5.



(a) *quiz solving*, Window 10, Low Overreliance



(b) *quiz solving*, Window 10, High Overreliance

Figure 7: Two Patterns in *quiz solving*, Windows 10 for Interpreting and Following LLM's responses.

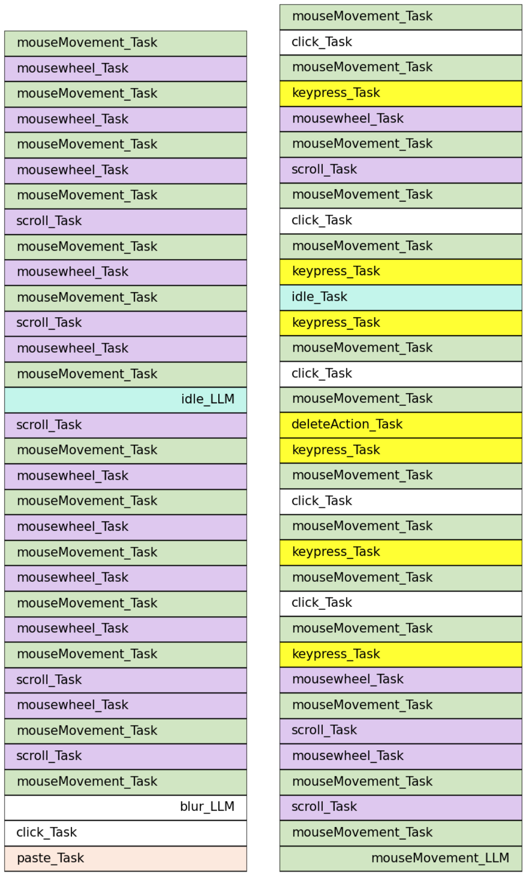
A.9 Explanation of Each Dimension for Action Vector

0–14: One-hot encoding for 15 action types.
15: Normalized timestamp.
16–34: Action-specific attributes (set to 0 if action event does not match):
- 21: totalMouseMovement (mouseMovement).
- 22: deltaY (mousewheel).
- 23: deltaY (scroll).
- 24: keyCount (keypress).
- 25: keyCount (delete).
- 26: textLength (copy).
- 27: textLength (paste).
- 28: textLength (highlight).
- 29–31: One-hot encoding for mousewheel direction (up, down, stationary).
- 32–34: One-hot encoding for scroll direction (up, down, stationary).

35–36: One-hot encoding for page type (AI page or tasksheet page).

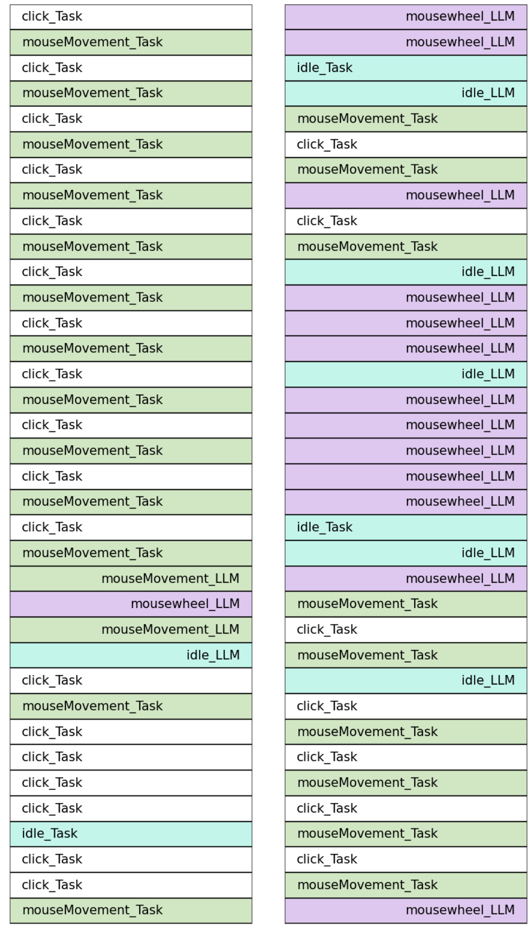
A.10 Computing Predictive Score for Test Action Sequence

Inspired by the regression approach in random forest algorithms [39], we designed a method to evaluate the model's predictive



(a) *article summarization*, Window 50, High Overreliance (Partial) **(b) *article summarization*, Window 60, Low Overreliance (Partial)**

Figure 8: Target Selection for *article summarization*



(a) *quiz solving*, Window 60, Low Overreliance (Partial) **(b) *quiz solving*, Window 30, High Overreliance**

Figure 9: Two Patterns in *quiz solving* (Supplement) for Interpreting and Following LLM's responses.

capability for overreliance. For each action sequence in the test set, we first computed its corresponding latent vector. Then, we found the top n ($n = 5$) nearest neighbor latent vectors in the training set that share the same cluster label. Finally, we averaged the actual overreliance scores of these training set neighbors to obtain the predicted score for the test action sequence.

A.11 Overreliance Levels

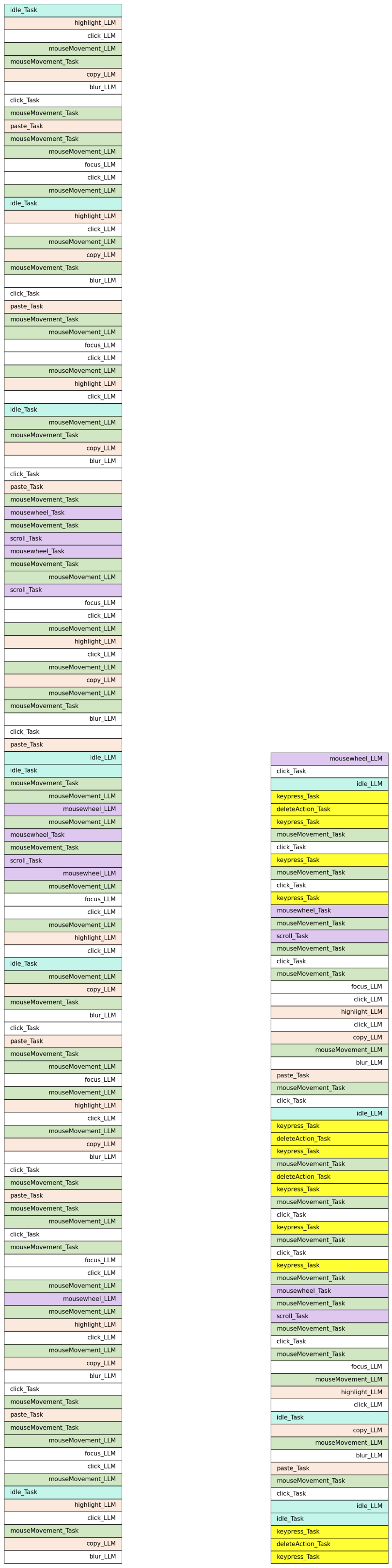
Lastly, to better characterize the overreliance scores of each cluster C_i, we compared the overreliance score of it to the rest of the data $D \setminus C_i$ (with D denote the full set of data) within each task-window combination. The results gives three categories:

(1) High overreliance: C_i's score significantly $> D \setminus C_i$
(2) Low overreliance: C_i's score significantly $< D \setminus C_i$
(3) Neutral: no significant difference

A.12 Questionaires after Each Task

Participants are required to rate from 1 to 7 for the following questions, with 1 indicating "not agree at all" and 7 indicating "extremely agree".

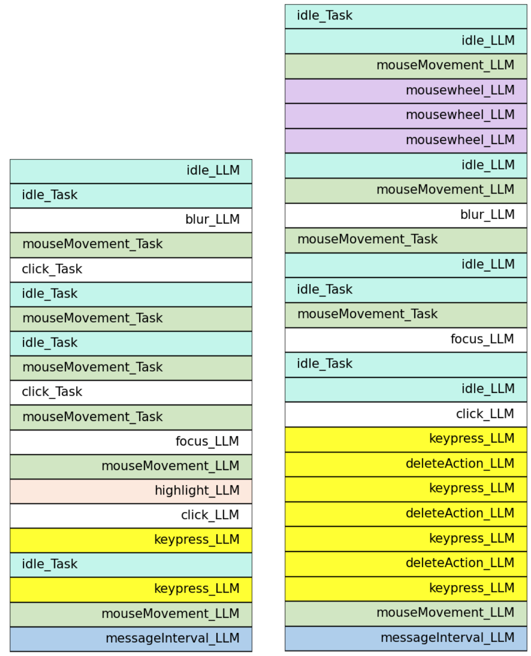
(1) The system is deceptive (words related to this: Deception, Lie, Falsity, Betray, Misleading, Phony, Cheat).
(2) The system behaves in an underhanded (sneaky, steal) manner.
(3) I am suspicious of the system's intent, action, or outputs (Mistrust, Suspicion, Distrust).
(4) I am wary (Beware) of the system.
(5) The system's actions will have a harmful or injurious outcome (Cruel, Harm).



(a) *trip planning*, Window 60, High Overreliance

(b) *trip planning*, Window 60, Low Overreliance

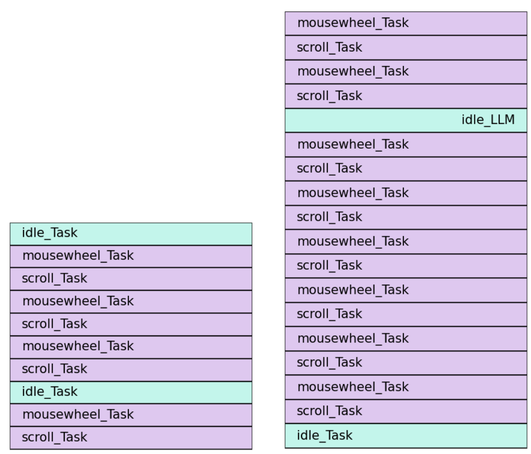
Figure 10: Copy and Paste for *trip planning*



(a) *quiz solving*, Window 50, High Overreliance

(b) *quiz solving*, Window 50, High Overreliance

Figure 11: Two Patterns in *quiz solving* Trusting LLM's responses



(a) *article summarization*, Window 10, Low Overreliance

(b) *article summarization*, Window 20, Low Overreliance

Figure 12: Consecutive Mousewheel for *article summarization*

(6) I am confident in the system (Assurance).

(7) The system provides security.

(8) The system has integrity (Honor).

(9) The system is dependable (Fidelity, Loyalty).

(10) The system is reliable (Honesty, Promise, Reliability, Trustworthy, Friendship, Love).

(11) I can trust the system.

(12) I am familiar with the system.

Participants are then required to answer the following question:

(1) In two sentences, explain how you arrived at your final answer. Describe how you utilized the information provided by LLM in reaching your conclusion.

Finally, participants are required to rate from 1 to 7 for the following questions, with 1 indicating "not familiar at all" and 7 indicating "extremely familiar".

(1) How familiar are you with the domain knowledge described in the task?

(2) How motivated are you to complete this task to the best of your ability?

A.13 Analysis Pipeline Details

This appendix provides additional technical details referenced in Section 4, including preprocessing procedures, feature encoding, model architecture, clustering configuration, and validation mechanisms used in the analysis pipeline.

A.13.1 Preprocessing Methodology. The raw interaction logs were first cleaned to exclude participants whose data were incomplete or corrupted due to upload issues or browsing errors. This yielded 62, 70, and 60 valid logs in *quiz solving*, *article summarization*, and *trip planning*, respectively. The remaining logs were merged across the Task Page and LLM Page by aligning all events chronologically using their start timestamps and labeling their page origin.

To eliminate noise and improve interpretability, we aggregated low-level events into higher-level action units based on temporal proximity and semantic similarity. For instance, multiple cursor movements within 200 milliseconds or continuous key presses were merged into a single action event. Each action was enriched with metadata including its duration, position on the page, and contextual type. Time was normalized such that the first observable action had a timestamp of 0.0, and the full 90-minute session duration was scaled to the range [0, 1.0]. Each action was assigned a unique sequential ID representing its temporal ordering within the session.

A.13.2 Action Vector Encoding. Each action sequence was transformed into a sequence of 37-dimensional vectors. These vectors included four main components: one-hot encoded action type (15 dimensions), a normalized timestamp (1 dimension), a one-hot encoded page context (Task Page or LLM Page, 2 dimensions), and a set of 19 additional features specific to the action type. These included continuous values such as cursor movement distance, scroll amount, prompt length, and burst typing duration, as well as categorical variables such as input modality or mouse wheel direction. Continuous attributes were log-transformed prior to training to mitigate skew caused by outliers. Categorical variables were encoded using one-hot or binary representations, depending on cardinality.

A complete list of encoded action attributes is summarized in Table 6. Vectorized sequences were used as direct input to the embedding model described in the next section.

Table 6: Summary of Encoded Action Features

Feature	Description	Number of Dimension
`action_type`	One of 15 categories (e.g., scroll, click, type, prompt-submit)	15
`timestamp`	Normalized session time (0.0 to 1.0)	1
`page_context`	Task page or LLM page	2
`total_mouse_movement`	Total distance moved by *mouseMovement* in physical pixels	1
`mouse_movement_duration`	The duration of a continuous *mouseMovement*	1
`scroll_duration`	The duration of a continuous *scroll* or *mouseWheel*	1
`scroll_distance`	Total distance moved by *scroll* in physical pixels	1
`mouseWheel_distance`	Total distance moved by *mouseWheel* in physical pixels	1
`scroll_direction`	*scroll* up, down or none	3
`mouseWheel_direction`	*mousewheel* up, down or none	3
`keypress_duration`	The duration of a continuous *keypress* input	1
`keypress_keyCount`	The total count of a continuous *keypress* input	1
`copy_textLength`	The total length of copied text	1
`paste_textLength`	The total length of pasted text	1
`highlight_textLength`	The total length of highlighted text	1
`delete_duration`	The duration of a continuous *delete* action	1
`delete_keyCount`	The total count of a continuous *delete* input	1
`idle_duration`	The duration of a continuous *idle*	1

A.13.3 Autoencoder Configuration. To generate a fixed-dimension representation of variable-length interaction sequences, we used a Transformer-based autoencoder model. Each preprocessed action sequence was prepended with a special `[CLS]` token initialized to a constant value. The encoder consisted of three Transformer layers, each with four self-attention heads, and intermediate feed-forward layers of size 128. The encoder output corresponding to the `[CLS]` token (size 64) was used as the final embedding of the entire sequence.

The decoder network mirrored the encoder structure, reconstructing the original sequence from the compressed embedding. Training was optimized with a combination of three loss terms: (1) categorical cross-entropy for recovering the original action type, (2) binary cross-entropy for page classification, and (3) a weighted reconstruction loss combining mean squared error (for continuous features) and cross-entropy (for categorical ones). Training used the Adam optimizer with a learning rate of 1e-4 and a batch size of 32. We trained a separate model for each task and window size, resulting in 18 autoencoders (3 tasks × 6 window sizes). Validation was conducted using a leave-one-participant-out setup to ensure generalizability, and models were early-stopped upon convergence.

A.13.4 Clustering Procedure. After embedding, each segmented sequence was represented by a 64-dimensional vector corresponding to the `[CLS]` output of the autoencoder. These latent vectors were clustered using the DBSCAN algorithm to discover commonly recurring patterns of behavior. To improve robustness and avoid parameter sensitivity, we performed multiple DBSCAN runs using a grid search over parameters: the neighborhood distance threshold (`eps`) was varied from 0.2 to 1.0 (step size 0.1), and the minimum required cluster size (`min_samples`) was varied from 3 to 10. Clusters that appeared in at least three different parameterizations were considered stable and retained for downstream analysis.

To assign clusters to unseen test points, we used a k-nearest neighbor classifier ($k = 5$) trained on the latent vectors and their corresponding cluster assignments from the training set. This allowed consistent cluster membership inference while preserving training/test separation.

A.13.5 Validation of Cluster Utility. After clustering, we validated each cluster using two criteria designed to ensure that the behavior it represented was both internally consistent and predictive of user overreliance.

First, for internal consistency (intrinsic similarity), we compared the distribution of overreliance scores for sequences in the training and test sets that had been assigned to the same cluster. We performed independent two-sample t-tests for each cluster. Clusters where the training and test scores were not significantly different ($p > .05$) were retained as stable across users.

Second, for predictive capability, we compared the mean overreliance score of the training sequences assigned to a cluster with the mean score of test sequences assigned to that cluster. We retained clusters in which the absolute difference between these two means was below a fixed threshold $\delta = 0.15$, empirically tuned to balance sensitivity with robustness.

A.13.6 Overreliance Score Stratification. To simplify interpretation, we discretized the continuous overreliance scores into three ordinal levels: high, neutral, and low. Users scoring in the top quartile within each task's overreliance distribution were labeled as high-overreliance, while those scoring in the bottom quartile were labeled as low-overreliance. Those with mid-range scores were labeled neutral. These labels were not used for inference or clustering but aided in interpreting cluster characteristics in qualitative analysis.

The methods described here provide the technical foundation for the findings reported in Section 5, where we qualitatively interpret representative clusters to shed light on behavioral patterns that align with LLM overreliance dynamics.

A.14 Pilot Study

Over time, we conducted multiple rounds of pilot studies with a total of 21 participants (9F, 12M). These pilots allowed us to testing different methods for capturing cognitive ground truth, calibrate task difficulty and constraints, and evaluate different categories of tasks.