GUIDE: A Benchmark for Understanding and Assisting Users in Open-Ended GUI Tasks

Supplementary Material

A. Detailed Evaluation Metrics

In this section, we provide the formal definitions for the evaluation metrics used across our four evaluation tasks: Behavior State Detection, Intent Prediction, Help Need Detection, and Help Content Prediction. Let N denote the total number of test samples in the dataset. For the i-th sample, let y_i denote the ground-truth label and $\hat{y}_i$ denote the model's predicted label. $\mathbb{I}(\cdot)$ denotes the indicator function, which equals 1 if the condition inside is true and 0 otherwise.

A.1. Metric Definitions by Task

A.1.1. Task 1: Behavior State Detection

This task is formulated as a multi-class classification problem where the model must classify a video segment into one of 9 distinct behavioral states. We evaluate performance using standard **Accuracy**.

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i = y_i) \tag{1}$$

A.1.2. Task 2: Intent Prediction

This task is framed as a Multiple-Choice Question (MCQ) task with 4 options (1 correct answer and 3 distractors). We use two metrics:

Accuracy. Measures the proportion of instances where the model selects the correct intent option from the four candidates.

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i = y_i) \tag{2}$$

Multi-Binary Accuracy (MBAcc). Following prior work [5, 8], we employ MBAcc to evaluate robustness against distractors. For a given sample i, let y_i be the correct option and $\mathcal{C}_i^- = \{c_{i,1}, c_{i,2}, c_{i,3}\}$ be the set of three incorrect distractor options. The model performs a pairwise comparison function $f(x, \text{opt}_A, \text{opt}_B)$ which returns the chosen option between A and B. A prediction is considered correct under MBAcc only if the model prefers the ground truth y_i over *every* distractor in $\mathcal{C}_i^-$.

$$\text{MBAcc} = \frac{1}{N}\sum_{i=1}^{N}\left(\prod_{c\in\mathcal{C}_i^-}\mathbb{I}(f(x_i, y_i, c) = y_i)\right) \tag{3}$$

A.1.3. Task 3-1: Help Prediction (Need Detection)

This sub-task is a binary classification problem (Help Needed vs. Not Needed). We evaluate this using Accuracy, Precision, Recall, and F1-Score. Let TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative) denote the classification counts.

- **Accuracy**: The ratio of correctly predicted observations to total observations.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

- **Precision**: The ratio of correctly predicted positive observations to the total predicted positives.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5}$$

- **Recall**: The ratio of correctly predicted positive observations to the all observations in the actual class.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6}$$

- **F1-Score**: The harmonic mean of Precision and Recall.

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

A.1.4. Task 3-2: Help Prediction (Content Prediction)

Similar to Intent Prediction, this sub-task is an MCQ task where the model must select the appropriate help content. It is evaluated using **Accuracy** and **Multi-Binary Accuracy (MBAcc)**.

Accuracy.

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i = y_i) \tag{8}$$

Multi-Binary Accuracy (MBAcc). Defined identically to the Intent Prediction task. Let $\mathcal{C}_i^-$ be the set of incorrect help content options for the i-th sample.

$$\text{MBAcc} = \frac{1}{N}\sum_{i=1}^{N}\left(\prod_{c\in\mathcal{C}_i^-}\mathbb{I}(f(x_i, y_i, c) = y_i)\right) \tag{9}$$

B. Dataset Details

We provide a comprehensive overview of the GUIDE dataset, detailing its statistical properties, task granularity, and the diverse range of software workflows it encompasses.

B.1. Dataset Statistics

GUIDE comprises a comprehensive collection of 120 screen recording videos, totaling approximately 67.5 hours of footage. A key characteristic of our dataset is the inclusion of rich verbal narration; as shown in Table B1, think-aloud narration covers **78% of the total video duration**, providing high-quality ground truth for annotating user intent and mental states.

Variable	Value
# Videos	120
Total Duration	67.5 hours
Avg. Duration	33 min 44 sec
Max Duration	1 hour 23 min 50 sec
Min Duration	16 min 42 sec
Think-Aloud Narration Ratio	78%
Task Samples & Granularity	
(1) Behavior State Detection	1.8K
Avg. Segment Length	14.16s
(2) Intent Prediction	1.3K
Avg. Segment Length	25.40s
(3) Help Prediction	1K
Avg. Segment Length	25.56s

Table B1. Statistics of the GUIDE dataset.

The dataset focuses on long-horizon, open-ended workflows. The average video duration is 33 minutes and 44 seconds, with sessions ranging from approximately 16 minutes to over 1 hour and 23 minutes (Figure B1). This extended duration ensures that the dataset captures the full evolution of user tasks, including periods of exploration, struggle, and error recovery.

Task Granularity. From these raw videos, we extracted varying numbers of instances for our three evaluation tasks. We collected **1.8K samples** for Behavior State Detection, **1.3K samples** for Intent Prediction, and **1K samples** for Help Prediction. Notably, the average segment length for behavior detection is shorter (14.16s) compared to Intent Prediction (25.4s) and Help Prediction (25.56s). This is because when annotating behavior states from narration-aligned segments, we instructed the model to split the clip if two or more states were identified.

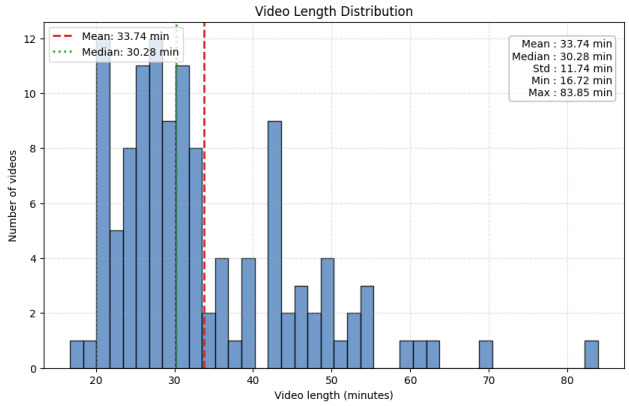


Figure B1. Distribution of screen recording video lengths in the dataset.

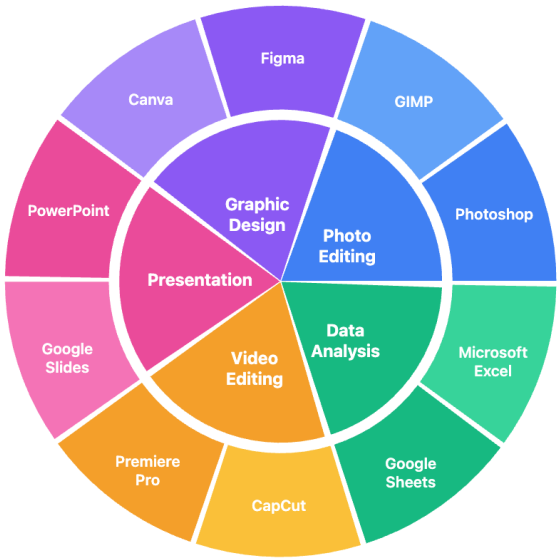


Figure B2. Software categories represented in the dataset.

Diversity. To ensure generalizability, the dataset spans a wide variety of software domains. As illustrated in Figure B2, users interacted with diverse applications ranging from creative design tools to analytical software.

B.2. Task Composition

To ensure our benchmark captures a comprehensive range of user behaviors, we designed a set of 20 open-ended tasks across five distinct software categories: Photo Editing, Graphic Design, Presentation Design, Video Editing, and Data Analysis. Table B2 provides a detailed overview of these categories and their corresponding tasks.

Open-Ended Task Design. Unlike rigid, step-by-step tutorials that result in linear behavior, our tasks are designed to be goal-oriented and open-ended. For instance, while we provided users with necessary materials (e.g., raw video clips, images) and suggested specific software features to utilize, we did not prescribe a fixed execution path or a target reference outcome. This semi-structured ambiguity is intentional; it forces users to engage in high-level planning, trial-and-error exploration, and problem-solving. Consequently, this setup naturally elicits the complex behavior states—such as *Exploration*, *Debugging*, and *Frustration*—that GUIDE aims to detect.

Domain Diversity. The selected software categories cover a broad spectrum of software domains, ensuring comprehensive coverage of diverse GUI workflows. Our dataset spans **creative domains** (Photo Editing, Graphic Design) that rely on visual manipulation and aesthetic decisions, **analytical domains** (Data Analysis) focused on data processing and logic, and hybrid tasks like **Presentation Design** or **Video Editing**. This variety ensures that our models are evaluated on their ability to generalize across diverse user interfaces, toolsets, and workflow paradigms.

Category	Software	Tasks
Photo Editing	Photoshop, GIMP	1. Create a composite from two images. 2. Create a bakery logo with a warm, friendly identity. 3. Replace a photo's background with a custom-designed pattern. 4. Design a movie poster.
Graphic Design	Figma, Canva	1. Design a mobile sign-up screen for a fictional app. 2. Design a custom 404 error page with a visual and animated element. 3. Design compact profile cards that display personal user details. 4. Design an event poster for a music festival.
Presentation Design	PowerPoint, Google Slides	1. Create a product pitch deck that highlights the MacBook's key features. 2. Create an interactive timeline presenting a company's history. 3. Create a 5-slide nature-themed shape-masked photo scrapbook. 4. Create a quiz deck with 3 multiple-choice questions.
Video Editing	Premiere Pro, CapCut	1. Edit a short interview to improve clarity and engagement. 2. Design a creative intro using animated text. 3. Edit a short instructional video to clearly guide a process. 4. Transform a long video into a highly engaging short-form clip.
Data Analysis	Microsoft Excel, Google Sheets	1. Design a Gantt chart for a mini project. 2. Summarize and visualize responses from a survey. 3. Visualize student performance across subjects. 4. Summarize and visualize product sales by category or region.

Table B2. Overview of open-ended tasks across software categories. Each category includes two software applications and four tasks designed to elicit natural and diverse user behaviors.

C. User Behavior Taxonomy

To effectively assist users, an agent must understand not just *what* the user is doing (e.g., clicking a mouse), but *why* they are doing it and what their current cognitive and behavior state is. We introduce a hierarchical taxonomy of 9 user behavior states, organized into four high-level phases of the software workflow: **Planning**, **Execution**, **Problem-Solving**, and **Evaluation**. Table C3 provides detailed definitions and examples for each state.

Behavior State	Description	Examples
Planning		
Task Understanding and Preparation	The user is focused on the logistics of the task. This includes interpreting the task, gathering necessary digital assets, and configuring the software environment. Their goal is to set up the conditions needed to begin the work.	Reading task instructions, opening required software/files/templates, arranging workspace (resizing windows, organizing directories), downloading images for photo editing.
Ideation and Planning	The user is engaged in high-level conceptual work. They are brainstorming ideas, outlining the structure of the outcome, or creating a plan for how to approach the task. This often involves creating preliminary, non-final content that serves as a guide.	Formulating high-level strategy, creating step lists, sketching rough layouts or wireframes. The output is a plan or outline, not the final polished product.
Execution		
Exploration and Decision-Making	The user experiments with different options or features to understand their effects and decide which one to use. This exploratory phase involves deliberate trial and comparison, often pausing forward progress to evaluate alternatives.	Applying effects and undoing them, hovering over tools to see what they do, testing multiple font sizes to decide which fits best.
Performing Actions	The user is confidently using the software to make progress on the task. These actions are purposeful and executed with little hesitation.	Typing/deleting text, inserting and resizing images, applying formatting with clear intent, searching for functions to use.
Problem-Solving		
Frustration	The user encounters a blocker and shows signs of being stuck, confused, or annoyed. The system may not behave as expected, or the user cannot find a way to perform a desired action, leading to repetitive or unproductive behavior.	Sighing, pausing for long periods, undoing repeatedly, clicking unresponsive elements, complaining about slow system behavior.
Debugging	The user moves beyond frustration and begins to actively investigate the cause of a problem. They form and test hypotheses to diagnose and fix an issue.	Testing alternative approaches, undoing recent actions step by step, forming hypotheses about causes, adjusting settings to identify errors.
Seeking External Help	The user recognizes a gap in their own knowledge and turns to an external resource for assistance or procedural guidance.	Switching to a web browser for solutions, opening tutorials/documentation, consulting AI assistants or colleagues, posting questions in forums.
Evaluation		
Waiting and Monitoring	The user is in a passive state, waiting for a system-controlled process to complete before continuing their work. They are unable to take meaningful action and typically observe progress indicators.	Watching loading bars or spinners, waiting for exports or rendering to complete.
Assessment	The user intentionally pauses their work to review and evaluate their output. They examine the result for quality, accuracy, or aesthetics.	Zooming in/out to inspect fine details, replaying video snippets for review, comparing results to reference images or previous versions.

Table C3. Taxonomy of user behavior states in open-ended GUI workflows.

C.1. Behavior State Distribution

Figure C3 illustrates the overall distribution of user behavior states across the four high-level phases defined in our taxonomy. Table C4 provides a granular breakdown of these states across the specific evaluation tasks.

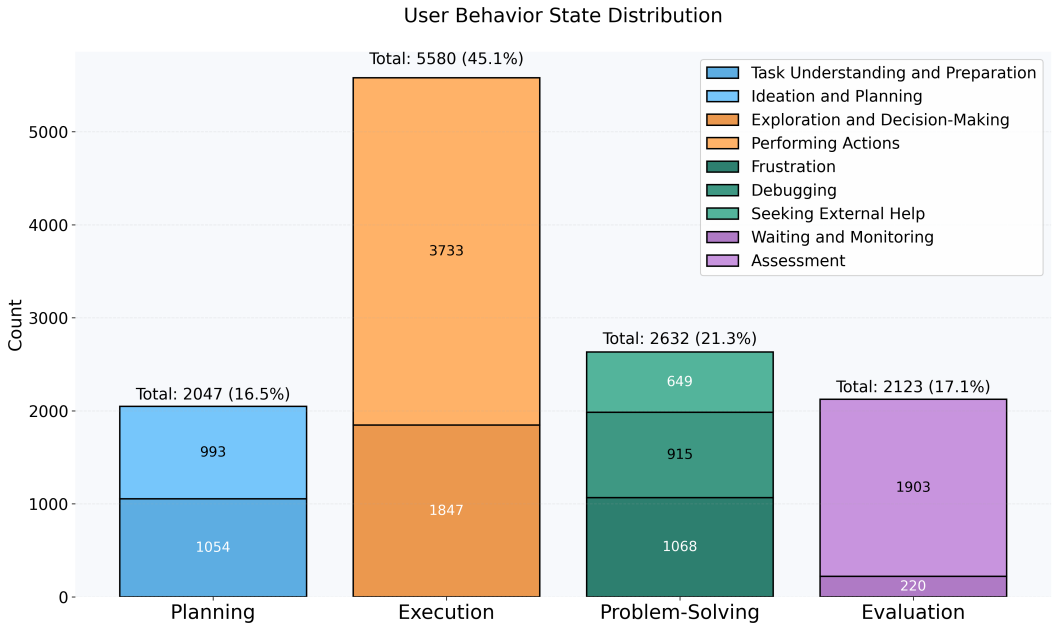


Figure C3. Distribution of user behavior states across Planning, Execution, Problem-Solving, and Evaluation phases across the videos in the dataset.

Table C4. Distribution of behavior state labels across the full dataset and specific evaluation tasks. Note that annotated instances used in the evaluation tasks may involve two or more states (e.g., a single segment containing both *Debugging* and *Seeking External Help*). Behavior State Detection uniformly sampled 200 instances from each class.

	Dataset		**Intent Prediction**		**Help Need Detection**		**Help Content Prediction**	
Behavior State	**Count**	**(%)**	**Count**	**(%)**	**Count**	**(%)**	**Count**	**(%)**
Planning								
Task Understanding and Preparation	1054	8.51%	216	9.85%	103	5.65%	36	2.82%
Ideation and Planning	993	8.02%	282	12.86%	84	4.61%	41	3.21%
Execution								
Exploration and Decision-Making	1847	14.92%	289	13.18%	162	8.89%	114	8.92%
Performing Actions	3733	30.15%	697	31.80%	474	26.00%	228	17.84%
Problem-Solving								
Frustration	1068	8.63%	131	5.98%	416	22.82%	415	32.47%
Debugging	915	7.39%	103	4.70%	259	14.21%	252	19.72%
Seeking External Help	649	5.24%	114	5.20%	85	4.66%	83	6.49%
Evaluation								
Waiting and Monitoring	220	1.78%	33	1.51%	17	0.93%	9	0.70%
Assessment	1903	15.37%	327	14.92%	223	12.23%	100	7.82%

C.2. Error Analysis: Behavior State Detection

Figure C4 presents the normalized confusion matrix for Behavior State Detection (Sec. 4.3.1). The results reveal a critical limitation in current MLLMs: a systemic bias toward interpreting interactions as productive execution while failing to recognize signs of struggle or hesitation. While models achieve reasonable accuracy for visually distinct states like *Seeking External Help* (0.61) and *Performing Actions* (0.57), they show near-zero capability in detecting *Frustration* (0.07) and *Debugging* (0.04). Instead, these negative states are overwhelmingly misclassified as *Performing Actions* (39% and 43%, respectively) or *Exploration and Decision-Making* (31% and 29%). This suggests that models perceive the visual activity of a struggling user—such as repeated clicking or rapid mouse movements—as deliberate progress, lacking the temporal understanding to distinguish between trial-and-error and confident execution.

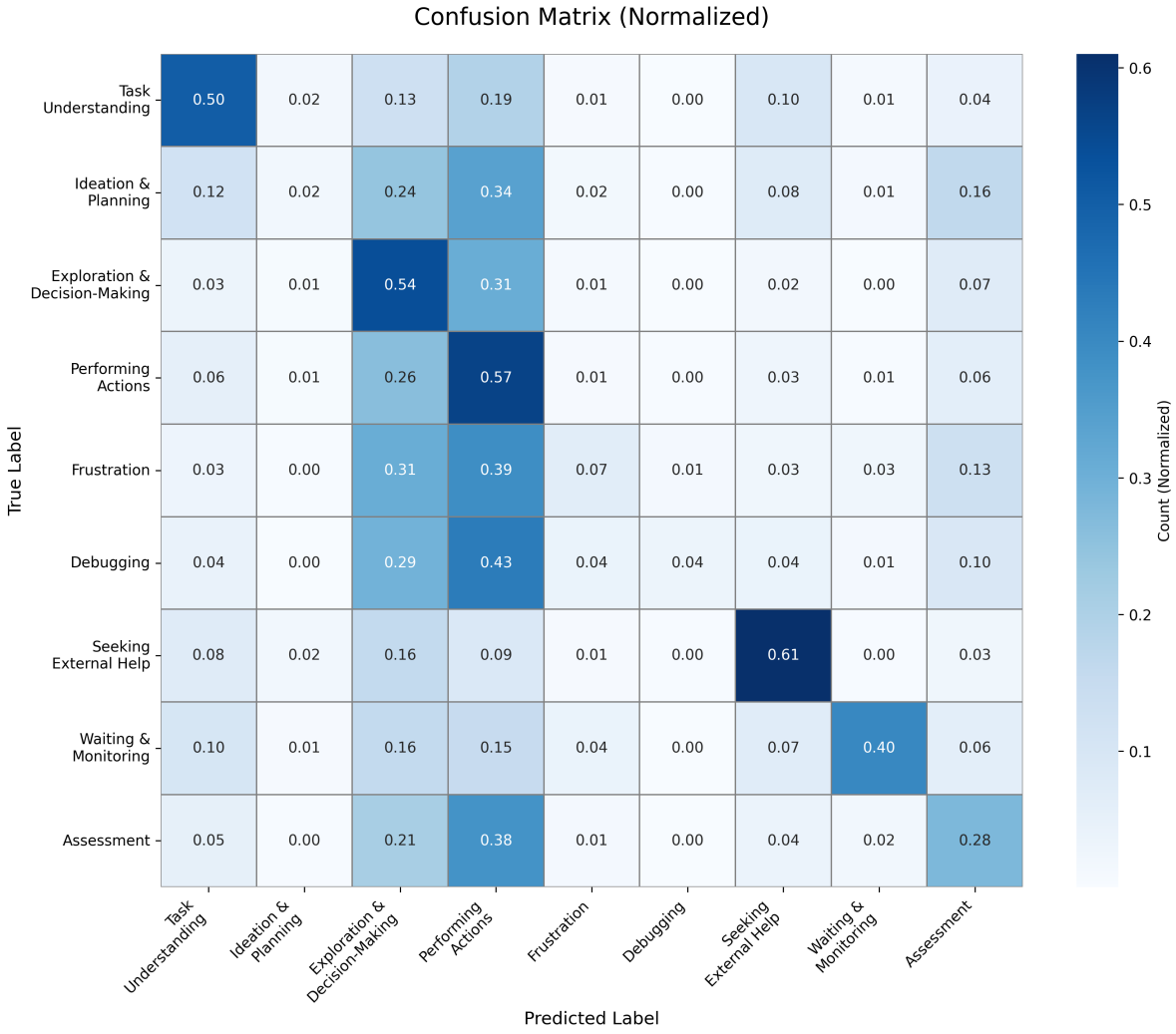
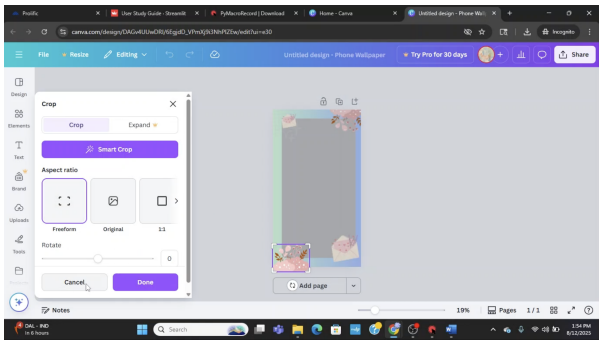


Figure C4. Normalized confusion matrix for user behavior state classification. The most common errors occur when *Frustration* or *Debugging* is misclassified as *Performing Actions* or *Exploration and Decision-Making*.

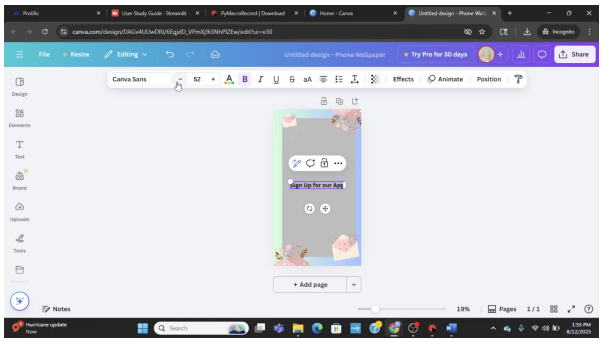
D. Screen Recording Video Examples

We present qualitative examples to illustrate the richness of the multimodal data in GUIDE.

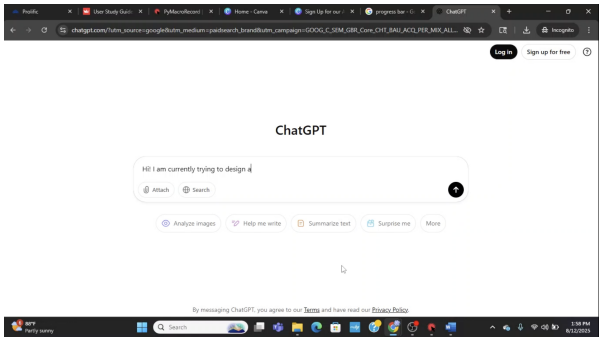
Software: Canva, **Task**: Design a mobile sign-up screen for a fictional app



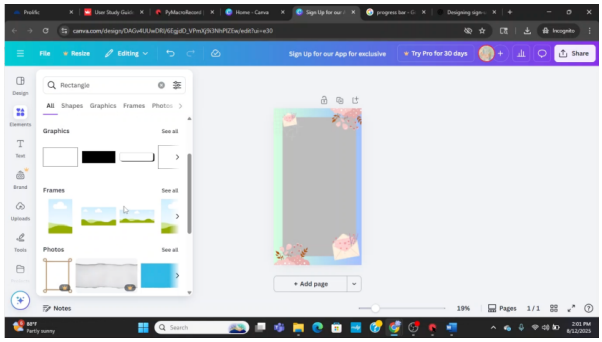
(4:11) "Nope, that's not what I wanted to do. Try again. Alright, let's do a text box."



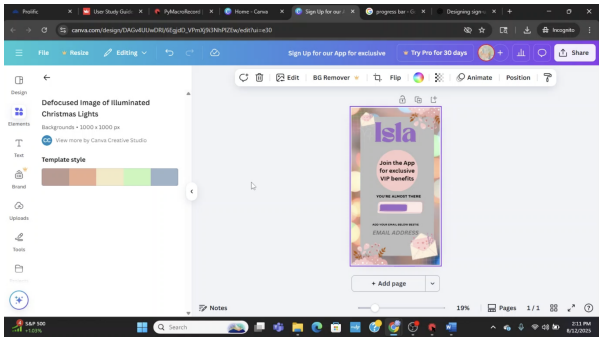
(4:40) "And we need to make this much smaller so it fits there at the top."



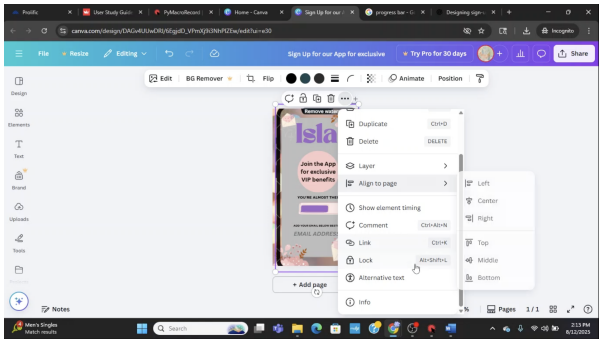
(6:50) "Progress bar for... what? I don't know. Just do progress bar. But you know what? Let's try ChatGPT because maybe they can help us."



(10:07) "Oh, that's cool, okay. So I can create the progress bar using the free elements with the shapes. Alright, so let's try and do that. Alright. Elements."



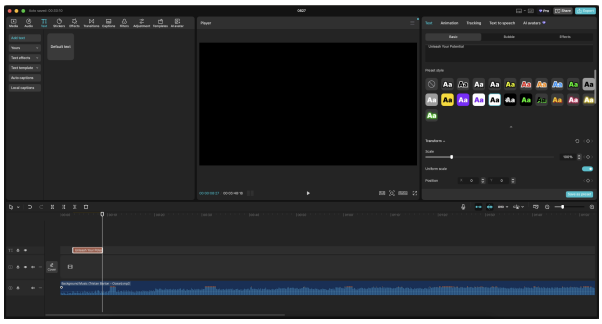
(21:01) "Ooh, what is this? Ooh, I like that so much better. All right, we're gonna keep this. Yes, we're gonna keep this."



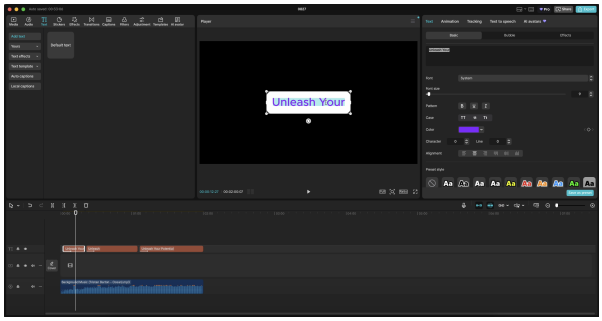
(22:05) "Not necessarily sure how I can... isn't there a way to like make it a certain size? How do I do that?"

Table D5. Example video illustrating the user's on-screen actions accompanied by think-aloud narration.

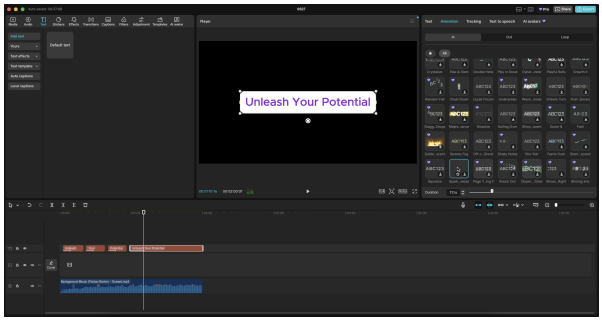
Software: CapCut, **Task**: Design a creative intro using animated text.



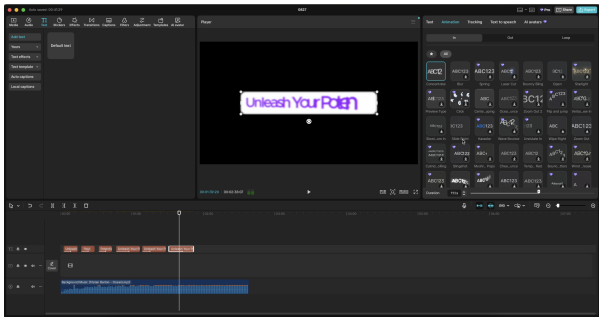
(4:56)"but I want to make it a bit more dynamic so all right."



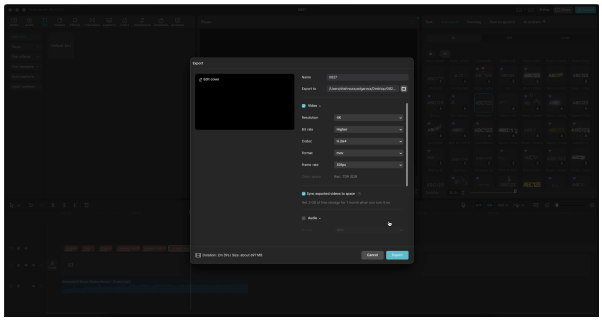
(7:55)"oh I think here it should be only one word appearing at a time."



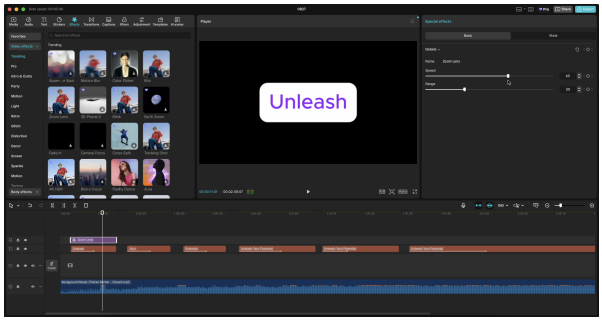
(11:39) "there are so many effects that I get a little bit too overwhelmed to see so many."



(17:55) "I would like to add additional different animation for this one so I like to keep this."



(18:42) "I am satisfied with the results so I will export this."



(20:14) "Maybe I would like to make it a bit slower."

Table D6. Example video illustrating the user's on-screen actions accompanied by think-aloud narration.

E. Benchmark Task Examples

E.1. Behavior State Detection

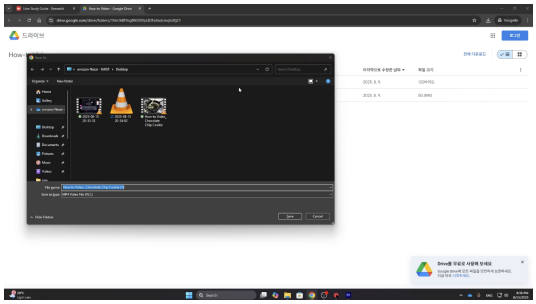
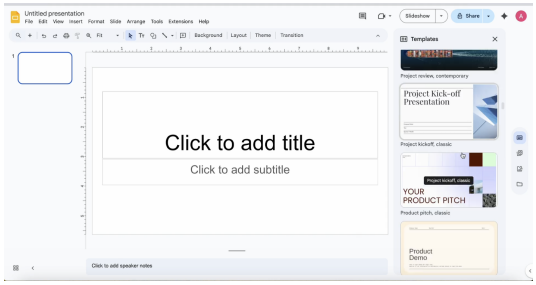
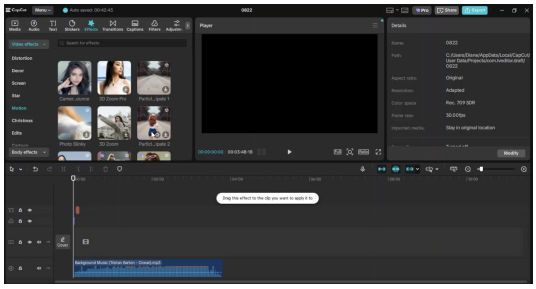
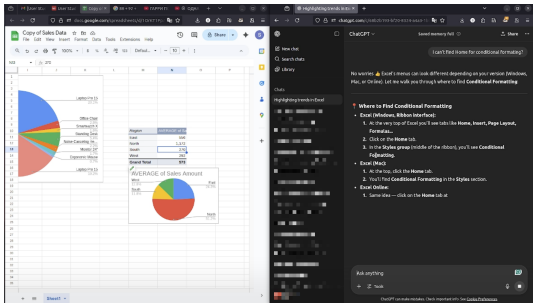
Screenshot	User Behavior State
 *"Okay, I downloaded it already. Delete my test, so I don't get confused. I have the video."*	**Software**: Premiere Pro **Task**: Edit a short instructional video to clearly guide a process. **Behavior State**: **Task Understanding and Preparation** The user is preparing their digital workspace before starting the editing task. They locate the necessary video file on their desktop and delete a superfluous 'test' file to prevent confusion.
 *"I would like to just use this design or the white some minimalistic like iOS design. Oh, this one. This one looks good. Okay, let's just..."*	**Software**: Google Slides **Task**: Create a product pitch deck highlighting a product's key features. **Behavior State**: **Exploration and Decision-Making** The user is actively browsing and comparing different templates, as shown by the scrolling and hovering behavior. The narration ('This one looks good') confirms they are evaluating options to make a final decision.
 *"Okay, that's strange. That's very strange, honestly."*	**Software**: CapCut **Task**: Design a creative intro using animated text. **Behavior State**: **Frustration** The user verbally expresses confusion ('that's strange') after the software behaved in an unexpected way. They are momentarily paused, indicating a blocker in their workflow before they decide on a new course of action.
 *(no narration)*	**Software**: Google Sheets **Task**: Summarize and visualize product sales by category or region. **Behavior State**: **Seeking External Help** The user is unable to find a feature and turns to ChatGPT for assistance. They type a question clarifying their problem, wait for the response, and then read the provided instructions.

Table E7. Example instances for the (1) User Behavior State Detection task, showing screenshots, think-aloud narration, and the corresponding behavior state.

E.2. Intent Prediction

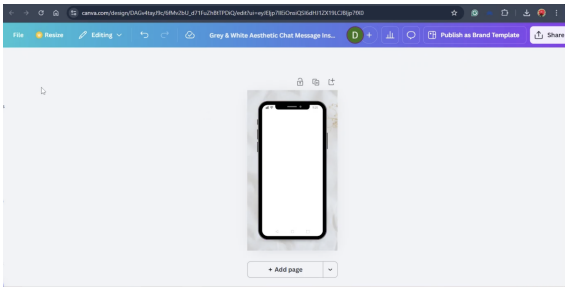
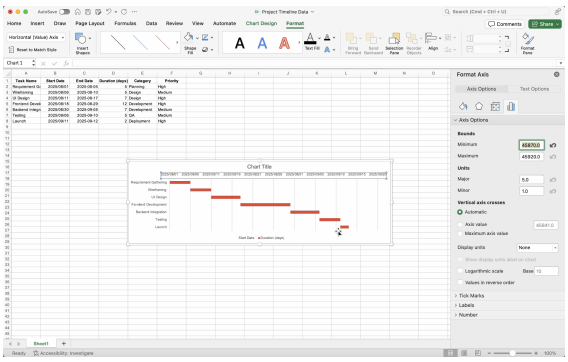
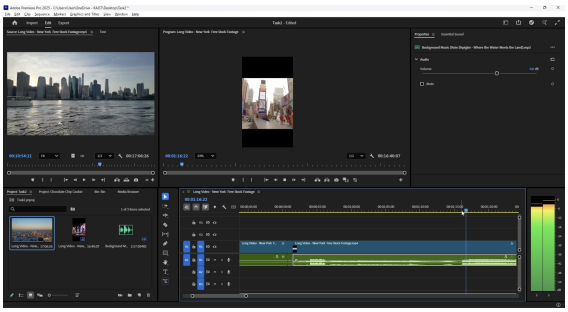
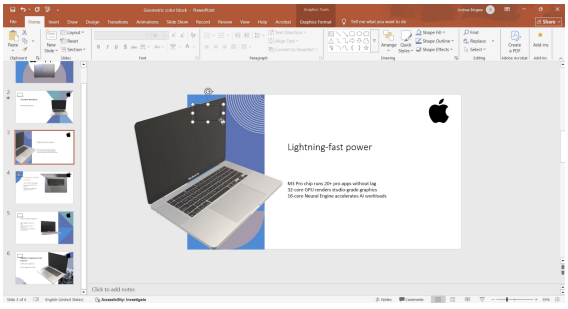
Screenshot	Intent
 *"So now that I have the frame as a design base, I need to include the input field for name, email."*	**Software**: Canva **Task**: Design a mobile sign-up screen for a fictional app. **Intent**: A: Rename the design file to reflect the new project **B: Add the required input fields to the design** C: Search for a suitable illustration to use as a header D: Resize the canvas to a custom dimension
 *"okay looks perfect, I need to adjust the end date as well"*	**Software**: Excel **Task**: Design a Gantt chart for a mini project. **Intent**: **A: Adjust the end date of the chart's horizontal axis** B: Adjust the date interval of the chart's horizontal axis C: Reverse the order of the chart's vertical axis D: Adjust the start date of the chart's horizontal axis
 *"When this slot comes, we should put some kind of image here."*	**Software**: Premiere Pro **Task**: Transform a long video into a short-form clip. **Intent**: A: Create a new text layer above the existing video track **B: Add an image to a specific empty slot in the timeline** C: Apply a transition effect to the end of a video clip D: Add a video clip to the end of the current sequence
 *"Paste, paste, paste, paste. Done. Done."*	**Software**: PowerPoint **Task**: Create a product pitch deck highlighting a product's key features. **Intent**: A: Align the logos with the main text boxes. B: Delete the logos from all the slides. **C: Duplicate the logos onto the remaining slides.** D: Change the color of the logos on all slides.

Table E8. Example instances for the (2) Intent Prediction task, showing screenshots, think-aloud narration, and the corresponding intent.

E.3. Help Prediction

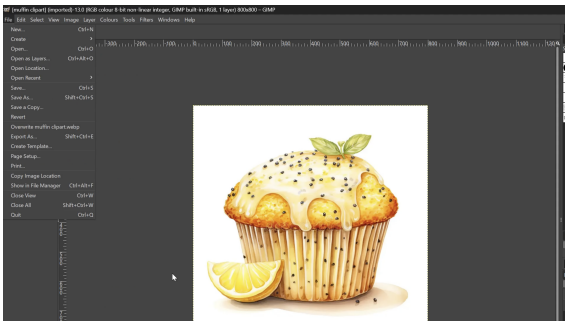
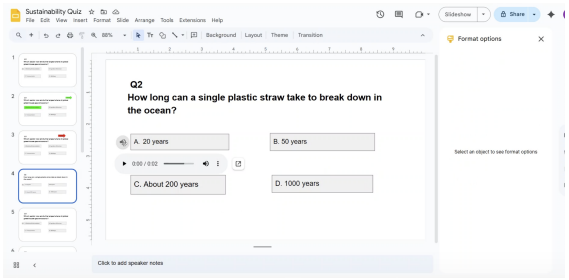
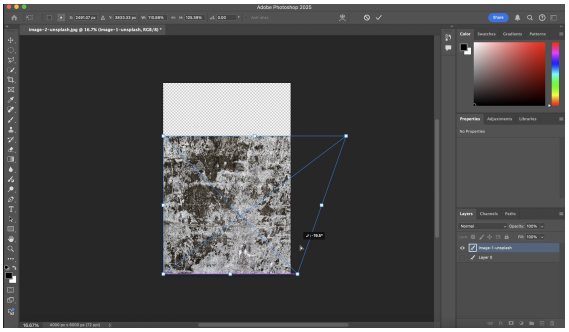
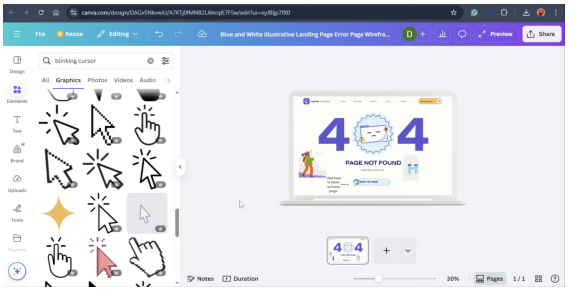
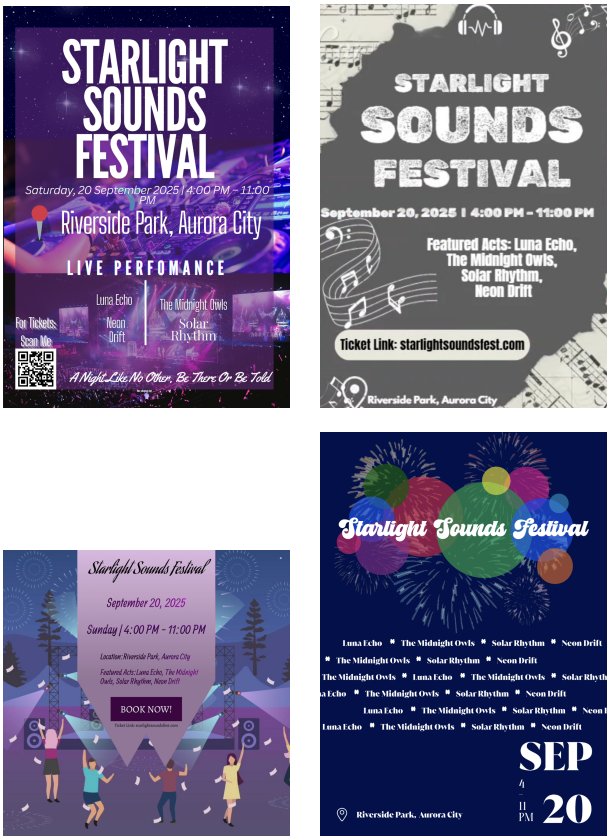
Screenshot	Help
 *"Where could I insert the text? [...] I'm just going to, because the help function I don't quite understand, but I can see if I can add it. Find it in Google."*	**Software**: GIMP **Task**: Create a bakery logo with a warm, friendly identity. **Help Content**: A: how to add another image as a layer **B: find the tool to add text** C: remove the image background D: add a background color or shape
 *"I think I made a mistake here and I need to rectify this."*	**Software**: Google Slides **Task**: Create a quiz deck with multiple-choice questions testing sustainability facts **Help Content**: A: align the answer choice boxes B: how to create a quiz slide template **C: how to fix a self-identified audio related error** D: add animation to reveal the correct answer
 *"I'll scale it. I just want to scale this up. How do I keep it?"*	**Software**: Photoshop **Task**: Create a composite from two images. **Help Content**: A: how to use the perspective or warp transform tools B: center the new layer on the canvas C: how to use layer blend modes **D: maintain aspect ratio while scaling**
 *"So I believe this is, this is great. I believe it's just simple."*	**Software**: Canva **Task**: Design a custom 404 error page with a visual and animated element. **Help Need**: A: help needed **B: no help needed**

Table E9. Example instances for the (3) Help Prediction task. For the Help Need Detection task, the top three instances illustrate cases labeled as *help needed*, while the last row shows an instance labeled as *no help needed.*

F. Software Task Outcome Examples

Figure F5 presents final artifacts produced by participants from the study. These examples highlight the open-ended nature of the assigned tasks. Despite receiving identical high-level instructions—such as "Design a poster for a music festival" or "Create a friendly bakery logo"—users produced markedly different results in terms of layout, aesthetic style, and complexity. This diversity confirms that the study elicited non-linear, creative workflows rather than fixed execution.



(a) Music event poster design in **Canva** (top) and **Figma** (bottom).

(b) Bakery logo design in **GIMP** (top) and **Photoshop** (bottom).

Figure F5. Example outcomes of the assigned tasks. The diversity across outputs reflects the open-ended nature of the tasks.

G. Human Verification Interface

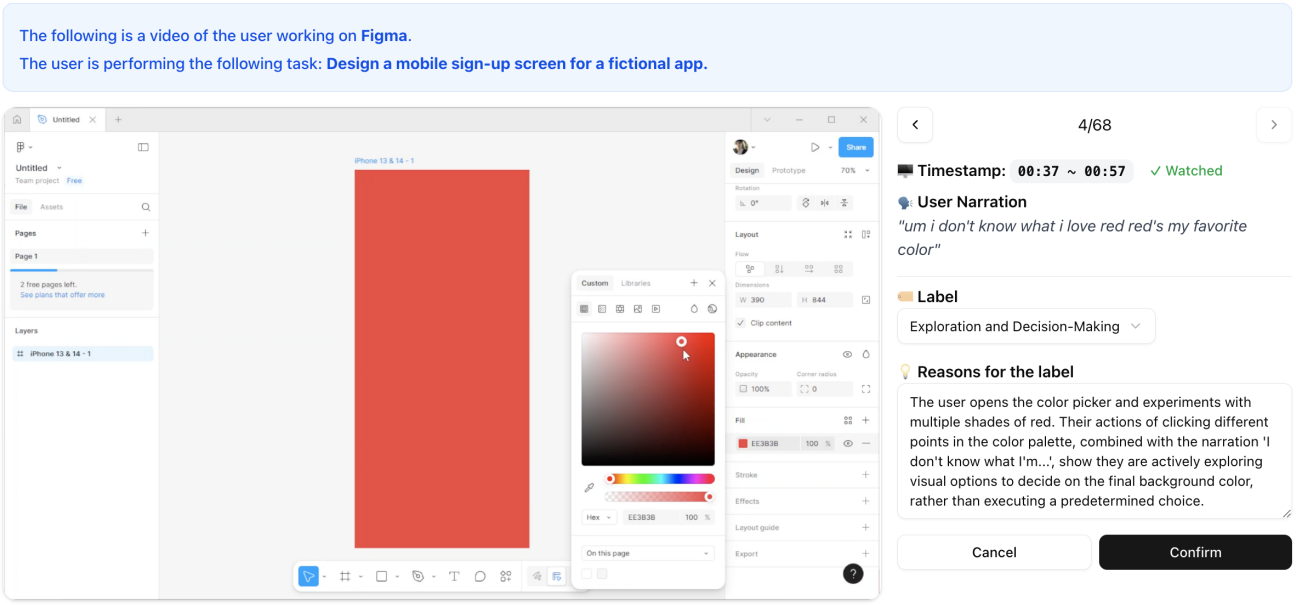


Figure G6. Annotation interface for validating and refining LLM-generated behavior-state labels. Annotators reviewed the predicted labels and the associated reasoning, correcting them if inaccurate. Each video segment was independently verified by two external annotators.

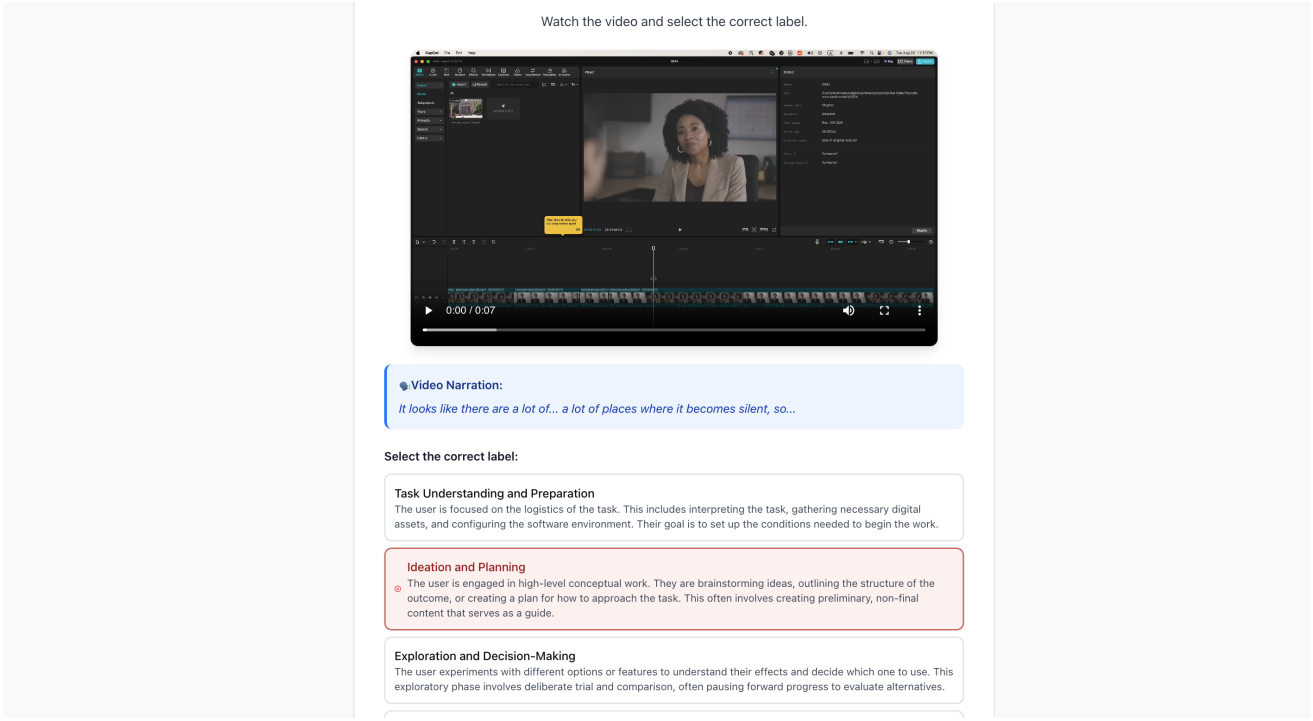


Figure G7. Before participating in the annotation, annotators completed a quiz phase where they had to correctly classify example video segments. This process ensured that all annotators possessed a solid understanding of the behavior taxonomy and definitions.

H. Prompts

H.1. Taxonomy of User Behavior State Generation

Taxonomy of User Behavior State Generation

```
# Goal: Create a comprehensive taxonomy of user mental and behavior states by analyzing the video recording
↪  and transcript. Integrate visual observations (screen interactions, UI changes, cursor behavior) with
↪  audio/verbal cues (tone, hesitations, verbal expressions) and transcript content (exact quotes, semantic
↪  meaning).

# Analysis Guidelines:
- VISUAL EVIDENCE: Describe what you see on screen (tool selections, menu interactions, visual feedback,
↪  cursor patterns)
- AUDIO EVIDENCE: Note tone of voice, hesitations, exclamations, and vocal expressions
- TRANSCRIPT EVIDENCE: Extract precise quotes that reveal mental states and intentions
- CROSS-REFERENCE: Connect visual actions with verbal expressions to understand user intent and mental state

# Output Format (return as JSON):
{{
    "taxonomy": [
        {{
            "label": "...",
            "definition": "...",
            "evidence": [
                {{
                    "timestamp": "00:01:32",
                    "modality": "visual",
                    "description": "User clicks on the brush tool and immediately switches to eraser",
                    "significance": "Indicates uncertainty or trial-and-error behavior"
                }},
                {{
                    "timestamp": "00:01:35",
                    "modality": "audio",
                    "description": "User says 'hmm, that's not right' with a frustrated tone",
                    "significance": "Verbal confirmation of confusion and frustration"
                }},
                {{
                    "timestamp": "00:01:35",
                    "modality": "transcript",
                    "quote": "hmm, that's not right, let me try something else",
                    "significance": "Shows problem-solving mindset and willingness to iterate"
                }}
            ]
        }},
        ...
    ]
}}

# Context:
Software: {SOFTWARE}
Task: {TASK_NAME}

# Transcript:
{TRANSCRIPT_JSON}

# Video Content:
SEE THE ATTACHED FILE.
```

Figure H8. Prompt to generate a taxonomy of user behavior states given demonstration videos.

H.2. Data Annotation

Behavior State Annotation

```
# Goal
You are given a screen recording video snippet of a user using the software {SOFTWARE}. Annotate the video
↪  with the provided taxonomy of user mental and behavior states. Include the taxonomy label and reasoning
↪  that explains why the label is appropriate for the video.

# Instructions
1. Your annotation label must be based on the user's current, on-screen behavior shown in the video.
2. Annotate video based on what you see, but if the label is not clear, use the think-aloud narration as
↪  auxiliary data to understand the user's intent or thought process.
3. Be aware that the user's narration may refer to past actions or future plans. Always align your annotation
↪  label with the user's current behavior at that specific moment in the video.

# Output Format (JSON)
{{
    "label": "..." // one of the labels in the taxonomy,
    "reasoning": "..." // Explanation on why the label is appropriate for the video,
}}

# Full Video Context
Software: {SOFTWARE}
Task the user is performing in the full screen recording video: {TASK_NAME}
The full transcript of the user: {TRANSCRIPT_JSON}

# Target Video Snippet Context
- Video snippet time range:
{TRANSCRIPT_JSON[narration_index]["start"]} - {TRANSCRIPT_JSON[narration_index]["end"]}
Narration sentences of the video snippet:
"{TRANSCRIPT_JSON[narration_index]["sentence"]}"

# Taxonomy
{TAXONOMY}

# Video Content
SEE THE ATTACHED FILE.
```

Figure H9. Prompt to annotate a given video segment based on the taxonomy of user behavior states.

Intent Annotation

```
You are analyzing a user's screen recording video and think-aloud narration while they use software (e.g.,
↪  video editing, design, or spreadsheet tools).

# Goal
For the given video snippet and its corresponding narration segment, infer what the user was aiming to
↪  complete or achieve by the end of this segment - their short-term goal or intention.

- The goal should represent **an outcome or result** that the user was either finishing or actively working on
↪  as the segment ends.
- Focus on **tangible, result-oriented goals** (e.g., "finish trimming the clip," "adjust the image color,"
↪  "complete text alignment").
- Ignore interface-level or purely procedural descriptions (e.g., "click this," "open that," "drag the
↪  layer").
- If no clear goal or outcome is expressed or shown, return "no tangible goal".

# Output Format (JSON)
{{
    "original_narration": "<verbatim narration text>",
    "goal": "<concise description of what the user aimed to complete or was completing by the end of this
    ↪  segment, or 'no tangible goal'>",
    "evidence_narration_snippet": "<the exact portion of the narration text that supports this inferred
    ↪  goal>",
    "reasoning": "<brief explanation of how this goal was inferred based on the narration and video>"
}}

# Full Video Context
Software: {SOFTWARE}
Task the user is performing in the full screen recording video: {TASK_NAME}
The full transcript of the user: {TRANSCRIPT_JSON}

# Target Video Snippet Context
- Video snippet time range: {TRANSCRIPT_JSON[narration_index]["start"]} -
↪  {TRANSCRIPT_JSON[narration_index]["end"]} seconds
- Narration during this snippet: "{TRANSCRIPT_JSON[narration_index]["sentence"]}"

# Video Content
SEE THE ATTACHED FILE.
```

Figure H10. Prompt used to annotate a given video segment with the user's intent.

Help Annotation

```
You are analyzing a user's screen recording video and think-aloud narration while they use software (e.g.,
↪  video editing, design, or spreadsheet tools).

# Goal
For the given video snippet and its corresponding narration segment, infer **what kind of help or guidance the
↪  user is looking for**.

- Focus on identifying explicit requests for help or moments where the user seeks information, clarification,
↪  or suggestions.
- Help-seeking can appear in two main forms:
  1) **On-Screen Help Behavior:** The user performs on-screen actions to seek help (e.g., opening a web
  ↪  browser, typing a query into a search engine or LLM, viewing online documentation or tutorials).
  2) **Narration-Based Help Expression:** The user verbally expresses confusion, uncertainty, or asks
  ↪  questions (e.g., "I don't know how to fix this," "Why is this not showing up?", "How do I do this?").
- Ignore casual statements or comments unrelated to problem-solving.
- If there is no clear indication that the user is seeking help, or if the type of help they need is implicit
↪  or ambiguous, return "no help needed".

# Output Format (JSON)
{{
    "help_needed": "<concise but meaningful description of the help sought - focus on the underlying need,
    ↪  such as 'explain masking feature', 'suggest alternative filter', 'debug export error', or 'clarify
    ↪  timeline snapping'",
    "help_source": "<'screen', 'narration', 'both', or 'none'>",
    "evidence_narration_snippet": "<exact portion of narration that indicates help-seeking (if any)>",
    "evidence_screen_behavior": "<description of what was seen on screen that indicates help-seeking (if
    ↪  any)>",
    "reasoning": "<brief explanation of how the need for help was inferred based on the narration or/and
    ↪  on-screen behavior>"
}}

# Full Video Context
Software: {SOFTWARE}
Task the user is performing in the full screen recording video: {TASK_NAME}
The full transcript of the user: {TRANSCRIPT_JSON}

# Target Video Snippet Context
- Video snippet time range: {TRANSCRIPT_JSON[narration_index]["start"]} {
↪  {TRANSCRIPT_JSON[narration_index]["end"]} seconds
- Narration during this snippet: "{TRANSCRIPT_JSON[narration_index]["sentence"]}"

# Video Content
SEE THE ATTACHED FILE.
```

Figure H11. Prompt used to annotate a given video segment with whether help is needed and, if so, what specific help is required.

Filtering On-Screen Help-Seeking Behavior

```
You are analyzing a user's sequence of help-seeking actions while they use {SOFTWARE} for the task
↪  "{TASK_NAME}".

You are given HELP_DATA, a list of chronological records where each record includes:
- index: original index of the record
- help: user's inferred help need (e.g., "find a graphic for a password field")
- screen_behavior: observed on-screen action
- narration: user's think-aloud narration

# Goal
- Keep only the help entries where the screen_behavior is about using external applications (e.g., Google
↪  Search, ChatGPT, Gemini, YouTube,etc.). Note that it doesn't include referring to the task instructions
↪  page.
- Return the list of segment ids that should be kept as a JSON object.

# Output Format (JSON)
Return only valid JSON:
{{
    "kept_segment_ids": [<int>, <int>, ...]
}}


# Data
HELP_DATA:
{HELP_JSON}
```

Figure H12. Prompt used to filter on-screen help-seeking behavior from segments previously marked as help needed.

Filtering Narration-Based Help-Seeking Behavior

```
You are analyzing a user's sequence of actions while they use {SOFTWARE} for the task "{TASK_NAME}".

You are given HELP_DATA, a list of chronological records where each record includes:
- index: original index of the record
- help: user's inferred help need (e.g., "find a graphic for a password field")
- screen_behavior: observed on-screen action
- narration: user's think-aloud narration

# Goal
Keep only segments where the narration explicitly asks for help or guidance.

# Definition of explicit help
The narration contains a direct request for help or instruction, for example:
- "help me", "i need help", "i want help", "can you help", "please help"
- How or where questions about operating the software, such as:
  "how do i ...", "how can i ...", "how to ...", "where is ...", "which option should i ...",
  "what should i click", "what does this do".
- Requests for instructions or explanation:
  "show me how to ...", "tell me how to ...", "could someone explain ...", "is there a way to ..."

# Exclude the following
- Exploration or intent without a help request: "i'm going to try", "let me see", "i will search"
- Trial and error or self-correction without a request: "no, not that", "okay now i got it", "ah ok",
↪  "finally"
- Uncertainty alone: "maybe", "i think", "not sure" unless followed by a direct question that asks for
↪  guidance
- Statements addressed to self that do not ask for help: "i need to add text", "i'm looking for an icon"
- Generic questions not tied to getting guidance on what to do next

# Output Format
Return only valid JSON:
{{
    "kept_segment_ids": [<int>, <int>, ...]
}}

# Data
HELP_DATA:
{HELP_JSON}
```

Figure H13. Prompt used to filter narration-based help-seeking behavior from segments previously marked as help needed.

Filtering No Help Needed

```
You are analyzing a user's sequence of actions while they use {SOFTWARE} for the task "{TASK_NAME}".

You are given NO_HELP_DATA, a chronological list of records where each record includes:
- index: original segment index
- start_time, end_time: time range of the segment
- no_help_reasoning: explanation for why the user does not need help
- narration: the user's think-aloud narration

# Goal
Identify up to 5 segments where it is **explicitly clear** that the user does not need help.
These are moments when the user:
- Performs actions confidently, smoothly, and intentionally.
- Demonstrates clear understanding of what to do next without hesitation or correction.
- Speaks in a calm, matter-of-fact tone (e.g., "Now I'll add text here," "Perfect," "That looks good.").

# Exclude the following:
- **Trial and error:** any sign of experimentation, correction, or rapid alternation (e.g., "No, no, no," "Let
↪  me try again," "Okay, that worked.")
- **Self-resolution after confusion:** phrases like "now I got it," 'finally," "oh, that's how," "ah okay," "I
↪  see," or any narration showing realization after failure or surprise.
- **Frustration or emotional reactions:** (e.g., "oh shit," "ugh," "why," "come on") even if followed by
↪  success.
- **Uncertain or exploratory speech:** "I think," "maybe," "let's see," "try," "not sure."
- **Segments that merely lack confusion** but do not clearly express confidence or mastery.

# Rules
- Select at most 5 segments that best reflect calm, deliberate, fluent progress.
- Preserve the original order of the selected segments.
- Return only the segment indices of the kept entries.

# Output Format (JSON)
Return only valid JSON:
{{
    "kept_segment_ids": [<int>, <int>, ...]
}}

# Data
NO_HELP_DATA:
{NO_HELP_JSON}
```

Figure H14. Prompt used to filter clear no-help-needed segments.

H.3. Model Evaluation

Behavior State Detection

```
# Goal
You are given a screen recording video snippet of a user working in {SOFTWARE}.
Classify the video into one of the labels from the provided taxonomy of user mental and behavioral states.
Include both the taxonomy label and a reasoning that explains why this label best fits the observed segment.

# Instructions
1. Base your classification on the user's on-screen behavior shown in the video.
2. Provide:
    - label: one of the taxonomy labels
    - reasoning: a concise explanation of why this label fits the segment
3. Return the output strictly in valid JSON, with keys "label" and "reasoning". Do NOT wrap the JSON in
↪  markdown code blocks. Return only the raw JSON object.

# Output Format (JSON)
{{
    "label": "...",
    "reasoning": "..."
}}


# Video Context
Software: {SOFTWARE}
Task performed in the full recording: {TASK_NAME}
Start and end times of the snippet (relative to the full recording):
{start_time} - {end_time} seconds

# Taxonomy Descriptions
{TAXONOMY}

# Previous Segment Context (*** Optional based on the condition)
The user behavior in the immediately preceding segment was {previous_label}: {label_definition}

# Video Content
SEE THE ATTACHED FILE.
```

Figure H15. Prompt used to evaluate the model on the (1) Behavior State Detection task.

Intent Prediction

```
# Goal
You are given a screen recording snippet of a user working in {SOFTWARE}.
Predict which option best describes the user's intention during this segment.

# Instructions
1. Watch the video and analyze on-screen actions.
2. Select the option (A-D) that best matches the goal of the user trying to achieve.
3. Use the provided behavior context to interpret the goal. (*** Optional based on the condition)
4. Return output in JSON:
   - label: one of A-D
   - reasoning: short explanation for your choice
5. Output only a valid JSON object (no Markdown).

# Output Format (JSON)
{"label": "A", "reasoning": "..."}

# Video Context
Software: {SOFTWARE}
Task performed in the full screen recording: {TASK_NAME}
Start and end times of the snippet (relative to the full recording):
{start_time} - {end_time} seconds

# User Behavior Context (*** Optional based on the condition)
The following user behavior is identified: {label}: {label_definition}.
Consider this context when predicting the user's intention and selecting the most appropriate option.

# Options
{options_text}

# Video Content
SEE THE ATTACHED FILE.
```

Figure H16. Prompt used to evaluate the model on the (2) Intent Prediction task.

Help Need Detection

```
# Goal
You are given a screen recording video snippet of a user working in {SOFTWARE}.
Based on the video content, the user's intention, and the user behavior context (*** Optional based on the
↪  condition), determine if the user needs help or not in this segment.

# Instructions
1. Watch the video and observe the user's behavior and actions.
2. Consider the user's intention and behavior context provided to better understand what they are trying to
↪  accomplish and their current state.  (*** Optional based on the condition)
3. Determine if the user needs help or not.
4. Provide:
    - label: "yes" if the user needs help, "no" if they do not need help
    - reasoning: a concise explanation of why the user does or does not need help based on the observed
    ↪  behavior.
5. Output only a valid JSON object (no Markdown).

# Output Format (JSON)
{{
    "label": "yes" | "no",
    "reasoning": "..."
}}

# Video Context
Software: {SOFTWARE}
Task performed in the full screen recording: {TASK_NAME}
Start and end times of the snippet (relative to the full recording):
{start_time} - {end_time} seconds

# User Behavior Context (*** Optional based on the condition)
The following user behavior is identified in order: {label}: {label_definition}.
Consider this behavior context when determining if the user needs help.

# User Intention (*** Optional based on the condition)
The user's intention or goal during this segment is: {intention}
Consider this intention when determining if the user needs help to achieve this goal.

# Video Content
SEE THE ATTACHED FILE.
```

Figure H17. Prompt used to evaluate the model on the (3-1) Help Need Detection task.

Help Content Detection

```
 # Goal
You are given a screen recording video snippet of a user working in {SOFTWARE}.
Based on the video content, the user's intention, and the user behavior context (*** Optional based on the
↪  condition), predict which option best describes what kind of help or guidance the user is looking for
↪  during this segment.

# Instructions
1. Watch the video and predict what help the user might need at this segment.
2. Consider the user's intention and behavior context provided to better understand what they are trying to
↪  accomplish and their current state. (*** Optional based on the condition)
3. Select the option (A, B, C, or D) that best matches what help the user needs to accomplish their intention
↪  given their behavior context.
4. Provide:
    - label: one of the option letters (A, B, C, or D)
    - reasoning: a concise explanation of why this option best fits the observed segment
5. Output only a valid JSON object (no Markdown).

# Output Format (JSON)
{{
    "label": "A",
    "reasoning": "..."
}}

# Video Context
Software: {SOFTWARE}
Task performed in the full screen recording: {TASK_NAME}
Start and end times of the snippet (relative to the full recording):
{start_time} - {end_time} seconds

# User Behavior Context (*** Optional based on the condition)
The following user behavior is identified in order: {label}: {label_definition}.
Consider this behavior context when predicting what help the user might need.

# User Intention (*** Optional based on the condition)
The user's intention or goal during this segment is: {intention}

# Options
{options_text}

# Video Content
SEE THE ATTACHED FILE.
```

Figure H18. Prompt used to evaluate the model on the (3-2) Help Content Prediction task.