Supplementary Material for VideoGUI: A Benchmark for GUI Automation from Instructional Videos

Contents

1 Experimental Settings **1**

1.1 Simulator Settings 1

1.2 Baseline Details 2

1.3 Evaluation Settings 2

1.4 Prompts Templates 4

2 Benchmark Statistics **9**

3 Simulator Experiments **11**

4 Qualitative Examples **14**

1 Experimental Settings

1.1 Simulator Settings

We use OBS Studio [1] software to record the demonstration videos and capture the user's screenshots. Notably, in the screenshots, the user's cursor is not recorded, which is beneficial as the screenshots can be used directly without revealing the target coordinates. We use `pynput` to monitor detailed user activity metadata, such as click location $[x, y]$, typed content, and scroll distance.

In Fig. 1, we display our manually labeled interface. Here, the annotator watches their key recording screenshots, with active regions such as the cursor coordinates highlighted in red. The annotators are then asked to enter the element name (e.g., "Drop-down menu of font color").

Submitted to the 38th Conference on Neural Information Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks. Do not distribute.

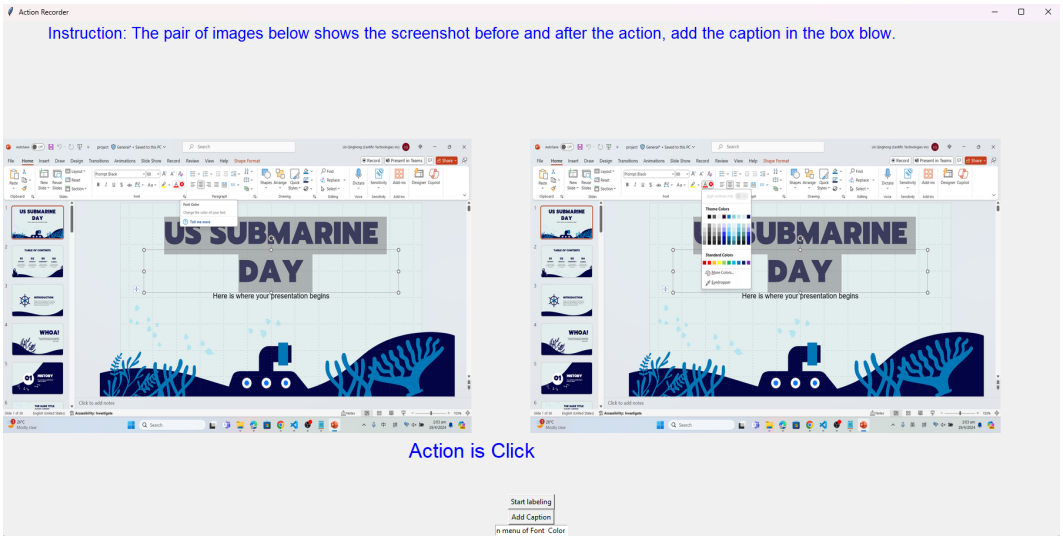


Figure 1: Illustration of Manual annotation tools. The user are asked to watch their keyframe in their recording, and prompt to provide the element name regarding action.

1.2 Baseline Details

Model	Ref. link	Version (*e.g.,* model id)
LLama3-70B [2]	deepinfra	`meta-llama/Meta-Llama-3-70B-Instruct`
Mixtral-8x22B [3]	deepinfra	`mistralai/Mixtral-8x22B-Instruct-v0.1`
GPT-3.5-Turbo [4]	OpenAI	`gpt-3.5-turbo`
CogAgent [5]	CogAgent	`CogAgent-18B`
Qwen-VL-Max [6]	Aliyun	`qwen-vl-max`
Claude-3-Opus [7]	Anthropic	`claude-3-opus-20240229`
Gemini-Pro-V [8]	Google	`gemini-pro-vision`
GPT-4-Turbo [9]	OpenAI	`gpt-4-turbo`
GPT-4o [9]	OpenAI	`gpt-4o`

1.3 Evaluation Settings

Click. We detail how we calculate the distance metric. Assume we have a ground-truth point $[x_o, y_o]$ while the screenshot size is $H \times W$.

• If the model prediction is a bounding box $[x_1, y_1, x_2, y_2]$ (*e.g.,*CogAgent [5] or Qwen-VL-Max [6]):

We cannot only take the center of the bounding box as the click target for evaluation because it does not account for the area of the bounding box. As illustrated in Fig. 2 (a), if the center point is very close to the ground truth but the bounding box cover a large area, the distance between the center point and the groundtruth would be small. Therefore, we design our metric to penalize for the area of the bounding box. Specifically, we calculate the distance between the ground truth and the four corners of the bounding box and then take the average. For the predicted bounding box, the average distance d is calculated as follows:

$$d = \frac{1}{4}\left(\sqrt{(x_o - x_1)^2 + (y_o - y_1)^2} + \sqrt{(x_o - x_1)^2 + (y_o - y_2)^2} + \sqrt{(x_o - x_2)^2 + (y_o - y_1)^2} + \sqrt{(x_o - x_2)^2 + (y_a - y_2)^2}\right)$$

• If the model prediction is a coordinate $[x_1, y_1]$ (*e.g.,*as in GPT4V+SoM [10]):

We directly adopt the distance d calculated by:

$$d = \sqrt{(x_o - x_1)^2 + (y_o - y_1)^2}$$

To normalize the pixel-level distance d to $0 - 1$, a simple way is to divide d by the maximum length in the screenshot, such as $\sqrt{H^2 + W^2}$. But in practice, the maximum length should be the distance

between the ground-truth point and the farthest vertices, so we use that for normalization. The comparison between the two normalization methods is illustrated in Fig. 2 (b).

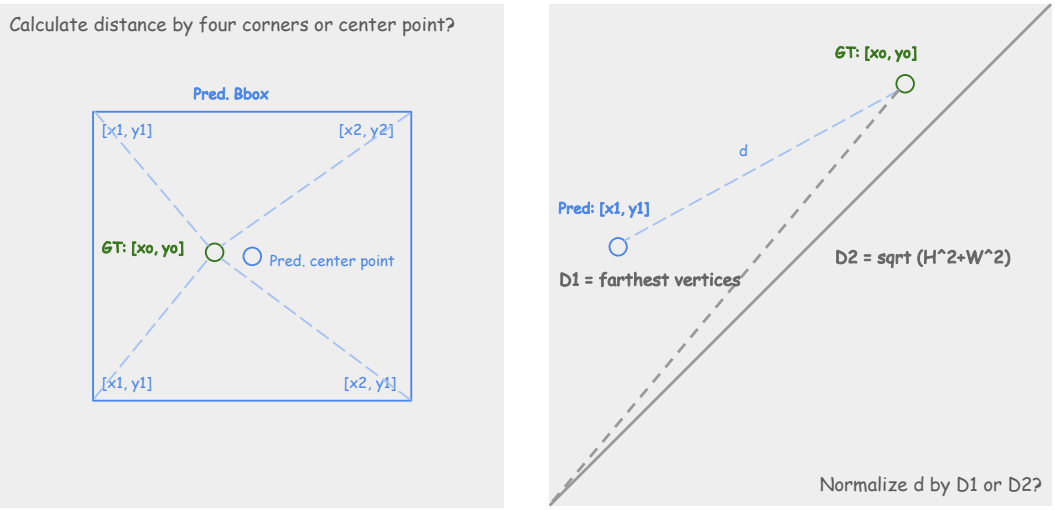


Figure 2: (a) Illustration of why taking the distance btween the center point of a bounding box and groundtruth is not a proper measure of model performance on click. As shown, the predicted bounding box center point is quite close to the ground-truth point, but the predicted bounding box area is large. **(b) Illustration of distance normalization.** To normalize the distance d to $0-1$, a more proper term should be $D1$ (farthest vertices) rather than $D2$.

Drag. Drag is a combination of Clicks, so we simply adopt the click metric for the start and end point of drag, and take the average. The score is calculated as Dist $:= \frac{1}{2}\left(\frac{d_s}{D_s} + \frac{d_e}{D_e}\right)$ where d_s is the pixel difference between predict start and GT start, while D_s is the farthest vertices for the GT start; d_e is the pixel difference between predict end and GT end, while D_e is the farthest vertices for the GT end;

For Recall, it is calculated by:

$$\text{Recall (start, end)} = \begin{cases} 1 & if \quad \text{Recall (start)} \,\&\, \text{Recall (end)} \\ 0 & otherwise \end{cases}$$

Type / Press. For type/press, we evaluates whether the model can generate correct and efficient code to control keyboard activity. First, we prompt LLMs to write code for typing activity, and then we use `pynput` to monitor the keyboard outputs by executing the code. In Fig. 3, we show the pipeline for evaluating type/press activity. The model must generate the correct actions (e.g., `Ctrl+F`) with high precision, avoiding unnecessary actions such as redundant `Ctrl` presses.

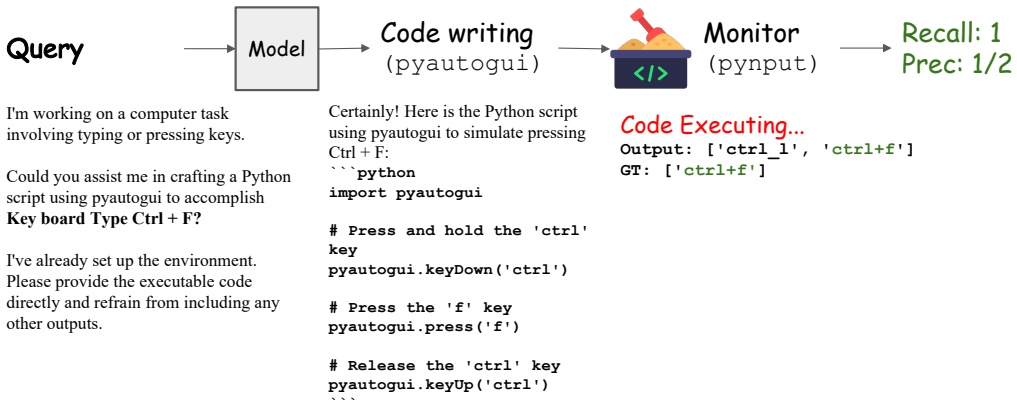
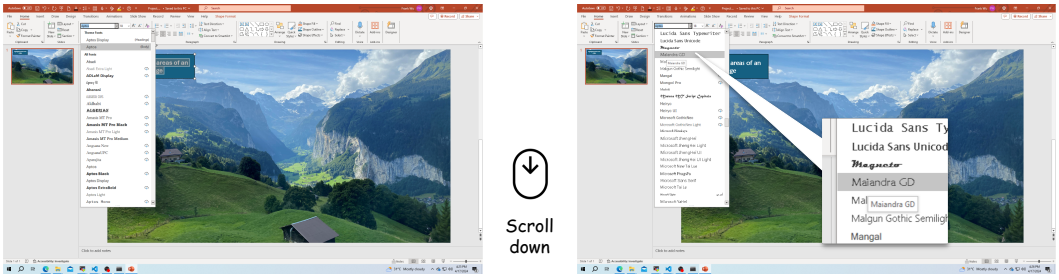


Figure 3: Illustration of how we evaluate the key / press action.

Scroll. Fig. 4 illustrates how we construction QA pairs to evaluate on scroll action. Before scrolling, the target element is assumed to be outside of the visible area, prompting for a scroll action. After

scrolling, the target element is assumed to be within the visible area, ready for the next action (*e.g.*,Click shown in the figure). Thereby, we can construct the QA pairs under these assumptions.



Before Scrolling, we assume that
[the target element] is **not within** the screenshot.

Question: Should I scroll to find the maiandra GD button?
Answer: Scroll down

After Scrolling, we assume that
[the target element] should **be inside** the screenshot.

Next Action: Click **[the maiandra GD button]**

Question: Should I scroll to find the maiandra GD button?
Answer: No

Figure 4: Illustration of how we create the scroll QA pair.

For each scroll, we create two QA pairs with the following GT answers: "scroll (up/down)" for the screenshot before scrolling and "no" for the screenshot after scrolling. We randomly shuffle the order of answer options to make the final testing samples.

1.4 Prompts Templates

Procedural Planning. In Tab.1 and Tab.2, we present the prompt templates for high-level and mid-level planning, respectively. These templates are conditioned on the query formulation, such as whether the start or end visual effects are provided, or paired with the textual query.

Action – Click. In Tab. 3, we show the template used by LLM to estimate click coordinates based on image resolution. With SoM's assistance, we use the Tab. 4 template to predict the mark index. With OCR's assistance, we use the Tab. 5 template.

Action – Drag. In Tab. 6, we show the template used by LLM to estimate drag coordinates based on image resolution. With SoM's assistance, we use the Tab. 7 template to predict the start and end mark index. With OCR's assistance, we use the Tab. 8 template.

Action – Type / Press. In Tab. 9, we present the template used by LLM to generate pyautogui code for keyboard actions.

Action – Scroll. In Tab. 10, we present the template used by LLM to predict scroll action, which is used for high-level planning. For mid-level planning, we remove the commentary component.

Evaluation. In Tab. 11, we display the evaluation template for GPT-4-Turbo [9].

```
def get_high_prompt(vis_start=True, vis_end=True, txt=None,
software=None):

  PROMPT = f"You are a software assistant professional at {software}."

  if vis_start and vis_end:
      PROMPT += "Given two sequence of image frames about the initial visual effect and
the final visual effect"
  elif vis_end:
      PROMPT += "Given a sequence of image frames about the final visual effect"
  else:
      PROMPT += " You are provided"

  if txt:
      PROMPT += " with a task textual description"

  PROMPT += " Your goal is to recognize the effect software demonstrates and pinpoint the
key functions or operations, necessary to replicate this distinctive pattern."

  PROMPT += """
**High-Level Planning**:
Distill the process into essential stages or components, emphasizing the unique functions
or operations, such as a specific design technique. Concentrate on brevity and precision in
describing each stage, highlighting the unique aspects that contribute to the overall effect.

Please format your response as follows (we use Powerpoint as an example):
‘‘‘
1: Insert a Circle and Change its color as black.
2: Add Text 'Happy' inside the Circle.
3: Apply the 'Fly-in' animation for the Circle.
’’’

Each stage should be concise yet comprehensive, focusing on the key functionalities or
operations that lead to the visual outcome in PowerPoint. Notably, avoid detailed step-by-step
actions. Strive to keep the number of stages as few as possible, only including those that are
crucial for achieving the unique effect.
"""

  if txt:
      PROMPT += f"**This is the textual descriptions** {txt}"
  return PROMPT
```

Table 1: High-level Planning Prompt conditioned on the interleaved instruction query.

```
def get_prompt(vis=True, txt=None, software=None):

  PROMPT = f"You have been assigned the task of planning a sequence of actions in {software} software to achieve a desired goal state based on certain conditions. Your objective is to outline the fundamental actions needed."

  if vis and not txt:
      PROMPT += "**You are provided with two screenshots which indicate the initial state as well as goal state.**"

  elif vis and txt:
      PROMPT += "**You are provided with a screenshot to indicate your initial state.**"

  if txt:
      PROMPT += f"**The goal is: {txt}**"

  PROMPT += """
Please format your response as follows:
‘‘‘
1. Click the ’xxx’.
2. Type ’yyy’.
3.: Click the ’zzz’.
’’’
Ensure that each step is clearly described to facilitate step-by-step reproduction of the actions.
"""
  return PROMPT
```

Table 2: Middle-level Planning Prompt conditioned on the interleaved instruction query.

I’m working on a computer task that involves clicking on some elements (like a button).
You are provided with a screenshot with a resolution of width: `{width}` and height: `{height}`.
Could you assist me in navigating to the "`{element}`"?
Please provide the location in the following format:
‘‘‘ [x, y] ’’’
Ensure that your response contains only the coordinates.

Table 3: Click action template that prompts LLMs output click’s coordinate [x,y]

The screenshot has been divided into areas and marked with numbers. Where is `{element}`?
Answer by mark index like [x].

Table 4: Click action template that prompts LLMs (with SoM [10]) output coordinate.

I’m working on a computer task that involves clicking on some elements (like a button).
Below are the OCR detection results (element name - bounding coordinates [[x1, y1], [x2, y2]]), which are separated by a colon ";".
`{ocr_result}`
Could you assist me in navigating to the "`{element}`"?
Please provide the location in the following format:
‘‘‘ [x, y] ’’’
Ensure that your response contains only the coordinates.

Table 5: Click action template that prompts LLMs (with OCR [11]) output click’s coordinate [x,y]

I am working on a computer task that involves dragging elements from one place to another
You are provided with a screenshot with a resolution of width: `{width}` and height: `{height}`.
Could you assist me in navigating for action "`{narration}`"?
Please provide the location in the following format:
“‘ [x1, y1] -> [x2, y2] ”’
where [x1, y1] are the start coordinates and [x2, y2] are the destination coordinates.
Ensure that your response contains only the coordinates.

Table 6: Drag action template that prompts LLMs output drag's coordinate [x1,y1] -> [x2, y2].

The screenshot has been divided into areas and marked with numbers.
To assist with dragging an item, please provide the start and end mark numbers.
How to `{element}`? Provide the mark indices as follows:
“‘ [x]->[y] ”’
where [x] represents the starting index and [y] represents the ending index.

Table 7: Drag action template that prompts LLMs (with SoM [10]) output SoM mark.

I am working on a computer task that involves dragging elements from one place to another
Below are the OCR detection results (element name - bounding coordinates [[x1, y1], [x2, y2]]), which are separated by a colon ";".
`{ocr_result}`
Could you assist me in navigating for action "narration"?
Please provide the location in the following format:
“‘ [x1, y1] -> [x2, y2] ”’
where [x1, y1] are the start coordinates and [x2, y2] are the destination coordinates. Ensure that your response contains only the coordinates.

Table 8: Drag action template that prompts LLMs (with OCR [11]) output drag's coordinate [x1,y1] -> [x2, y2].

I'm working on a computer task involving typing or pressing keys.
Could you assist me in crafting a Python script using pyautogui to accomplish `{goal}`? where the key input element is "`{element}`".
I've already set up the environment.
Please provide the executable code directly and refrain from including other outputs or additional code blocks. Ensure that your response contains only one code block formatted as follows:
“‘python
import pyautogui
pyautogui.press('ctrl')
“‘

Table 9: Type / Press action template that prompts LLMs output `pyautogui` code.

I'm currently engaged in a computer-based task and need your assistance.
You are provided with an image of my screenshot.
Could you advise whether I need to scroll to see the complete element "`{element}`"? Please note that even if the element appears partially, I still need to scroll to see it completely.

'A': 'No need to scroll.', 'B': 'Scroll down.', 'C': 'Scroll up.'

Please select the appropriate option and format your response as follows (Wrap options in square brackets):
"' [A] "'

Notably, only output options with square brackets

Table 10: Scroll action template that prompts LLMs to output a decision like scrolling (up/down) or not.

You are tasked with evaluating the quality of a software procedure plan. Assess the prediction provided by an AI model against the human-generated ground truth and assign a correctness score to the prediction.
Evaluation Criteria:
1. *Conciseness and Clarity*: The procedure plan should be straightforward and to the point.
2. *Element Accuracy*: Pay attention to the precision of specific details like types of animation, text content, and design elements (e.g., 3d shape, color, shape). The prediction should accurately reflect these aspects as mentioned in the ground truth.
3. *Commentary*: Provide a brief commentary in your response summarizing the accurate and inaccurate aspects of the prediction as evidence to support your scoring decision.

Correctness Score (must be an integer):
- 0: Completely incorrect
- 1 to 3: Partially correct (with 1 being least accurate and 3 being more accurate)
- 4 to 5: Fully correct (with 4 being good and 5 being perfect)

Ground truth:
`{GT}`

Prediction:
`{Pred}`

Considering the detailed elements and the overall process, please format your response as follows:
```
[comment]:  Summary of evaluation.
[score]:  x
```

Table 11: Evaluation Prompt Template

2 Benchmark Statistics

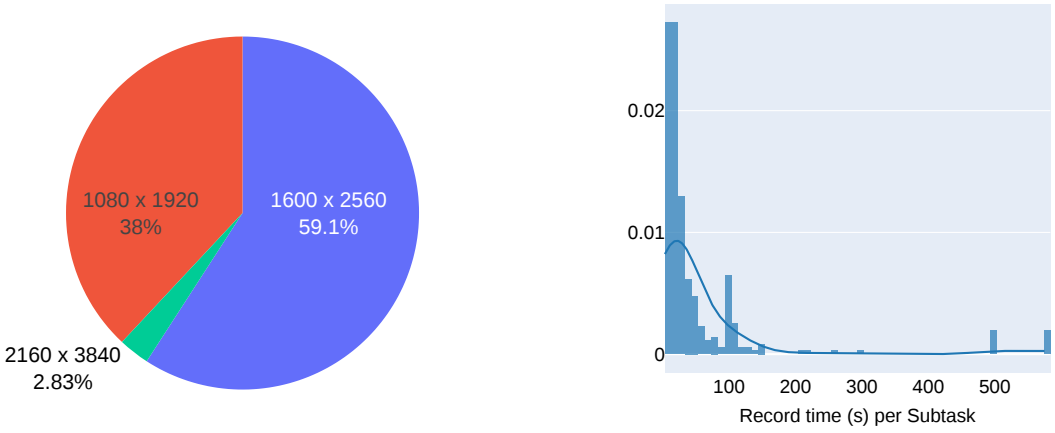
Software distributions In Tab. 12, we present the software distribution on VideoGUI.

Software	Platform	# Full Task	# Subtask	# Action per full task	# Action per subtask
Powerpoint	Windows	8	52	47.6	8.5
StableDiffusion	Web + Windows	10	69	19.0	4.0
Runway	Web	11	63	24.3	4.7
Photoshop	Windows	10	69	19.0	4.0
After Effects	Windows	13	67	29.3	7.2
Premiere Pro	Windows	7	38	15.4	4.5
Capcut	Web + Windows	10	46	9.4	3.6
DaVinci	Windows	11	44	18.8	4.7
YouTube	Web	0	13	0	4.3
Web Stock	Web	0	12	0	9.7
VLC player	Windows	0	12	0	9.2
Total	–	82	463	23.7	5.8

Table 12: VideoGUI's software distribution.

Manual Recording Cost. In Fig. 5a, we present the screenshot resolution distribution primarily used for action execution.

Screenshot's resolutions. In Fig. 5b, we present the distribution of manual recording time per subtask, with an average of 55 sec.



(a) Screenshot resolution distribution.

(b) Recording duration per subtask.

Figure 5: Distribution of **(a) Screenshot resolution** and **(b) Human recording time.**

World Cloud. In Fig.6, we present VideoGUI's Word Cloud, where the most frequent words are atomic actions (*e.g.,* click, drag, type) and commonly used proper nouns (*e.g.,*, layer, background, pannel) in the GUI.

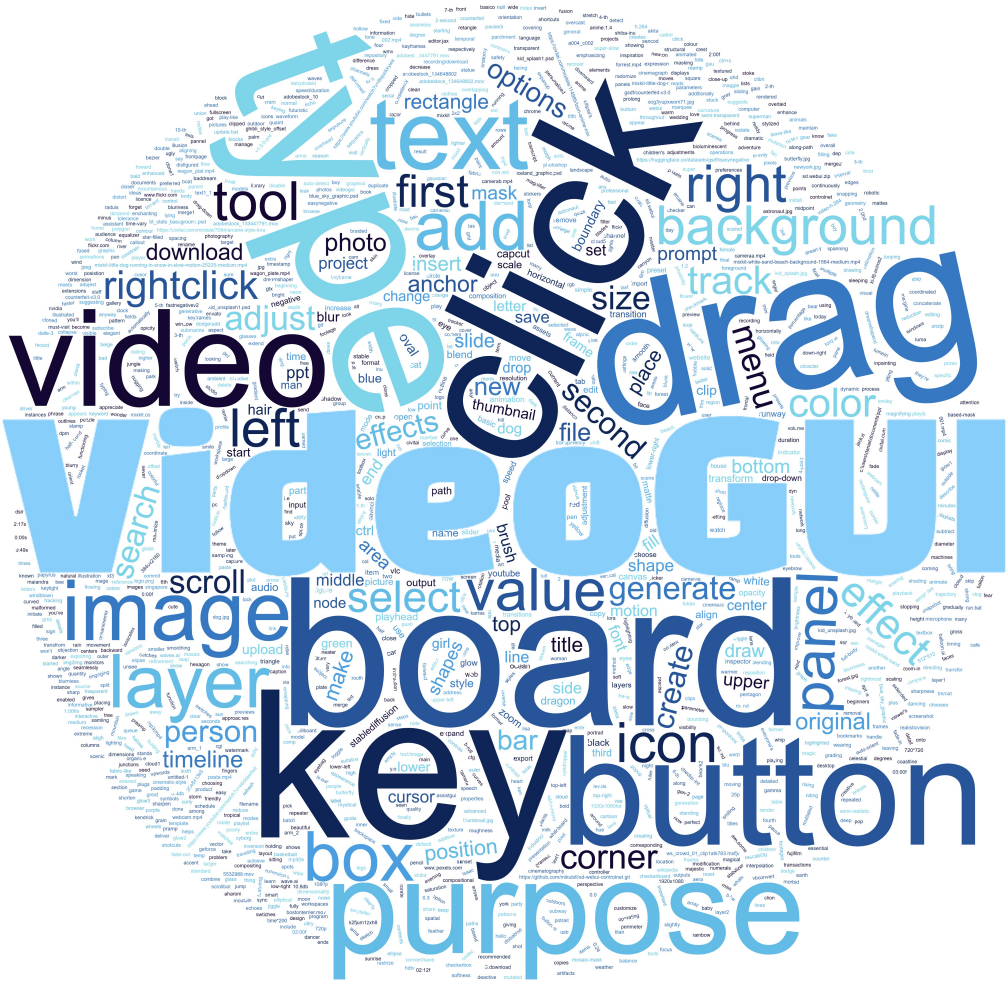


Figure 6: VideoGUI World Clouds

3 Simulator Experiments

Real-world Simulator. To simulate the real application scenario, we use the best performing LLM GPT-4o and build a simple agent baseline as shown in Fig. 7. We evaluate this agent on the most popular software (Powerpoint) to study its behavior.

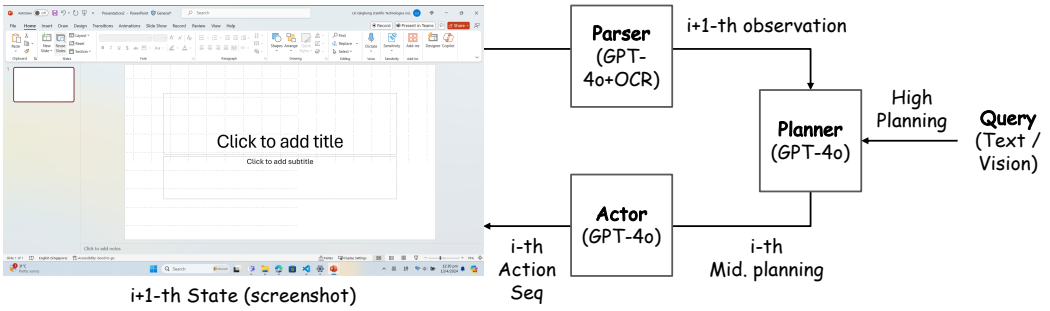


Figure 7: Our Minimalist GUI Agent Framework consists of three components: a Parser, a Planner, and an Actor. The Planner receives input queries, which may be either vision previews or text instructions. It then conducts high-level planning and generates mid-level plans for the Actor. The Actor executes these plans by performing a sequence of actions. After action execution, the current state (screenshot) is captured and sent back to the Parser to gather observations. These observations are then relayed to the Planner for subsequent planning.

Model	Settings	VideoGUI Eval.			Full task Eval.	
		High Plan.	Mid Plan.	Action	Success Rate	Rank (Arena) ↓
GPT-4o [9]	Orig. Query (V)	17.1	53.5	56.3	0	2.50
	w. GT High Plan.	100.0	53.5	56.3	0	1.88
	w. GT High & Mid Plan.	100.0	100.0	56.3	0	**1.38**

Table 13: Simulator Evaluation on VideoGUI's PPT *full tasks*.

Tab. 13 presents the model performance on full task execution in our simulator environment. We see that completing the full task is extremely challenging for the GPT4o agent, with a notable 0 success rate for all variants. This again supports the design of our hierarchical evaluation, as the zero success rate simply implies the model/agent fail to execute the full task, without enough information in where they succeed or fail, or even how these models/agents perform relatively to each other. Therefore, we introduce another metric, Rank (Arena), which compares the final outcome of their execution. Specifically, we ask human to perform manual inspection, and rank the comparing models by the similarities between the final results and the GT. We found that when injected with GT planning (both high or mid.-level), the full-task execution can be significantly improved. These results echoes our observations of low model performance in high-level and mid-level planning in the main paper, which are the bottlenecks of successful full-task executions.

We visualize the final outcome of the three agent variants in Fig. 9 and Fig. 11.

Additionally, in our Supplementary, we have included **human demonstration videos** alongside **agent execution videos** for these two samples. Observing them reveals that the GUI agent remains very slow and struggles to complete each task smoothly.

Model	Settings	VideoGUI Eval.		Subtask Eval.	
		Mid Plan.	Action	Success Rate (%)	Avg. Round ↓
GPT-4o [9]	Orig. Query (V+T)	53.5	56.3	20.0	5.4
	w. GT Mid Plan.	100	56.3	**50.0**	**3.3**

Table 14: Simulator Evaluation on VideoGUI's PPT *subtasks*.

In Tab. 14, we examine the performance of the GPT-4o agent in subtask competitions. Since subtasks do not necessitate high-level planning, we primarily investigate two variants: one with and one without

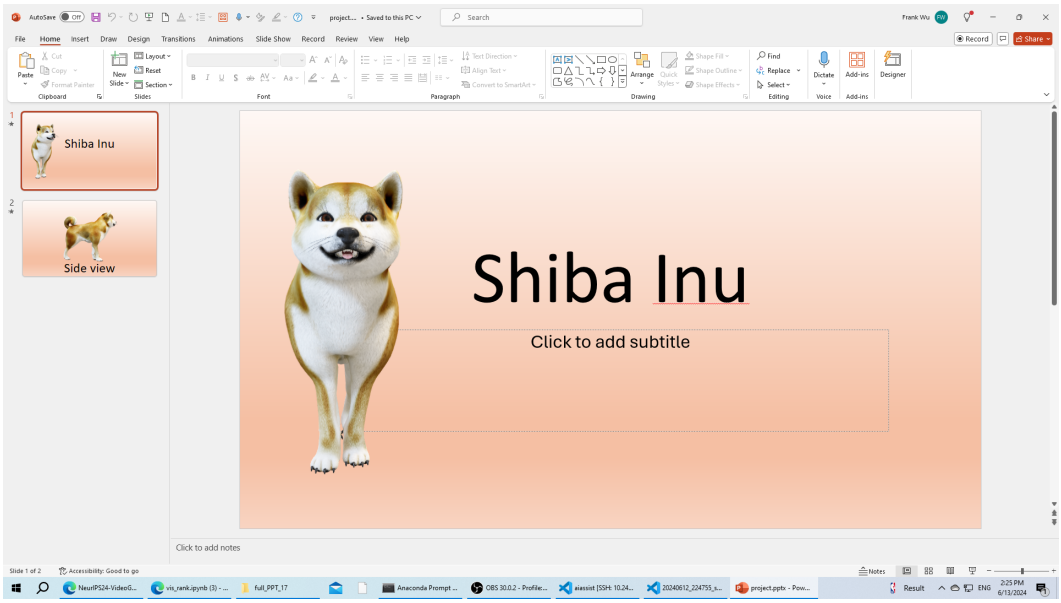
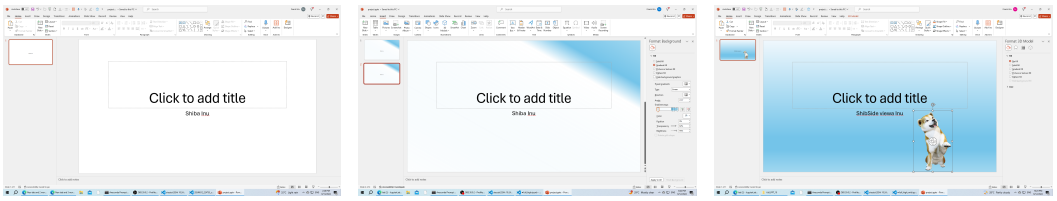


Figure 8: Final effect in Powerpoint files.



(a) GPT-4o **(b) GPT-4o w. GT High Plan** **(c) GPT-4o w. GT High+Mid. Plan**

Figure 9: Example of final outcome with our simple GPT-4o agent in simulated environment. When provided with GT planning (c), the GUI agent successfully inserts the 3D model. However, it still fails to match the background color.

manually provided middle-level planning, referred to as action sequences. Our study yields two key findings: (*i*) Despite the simplicity of these tasks, the original GPT-4o agent achieves a success rate of only 20.0%. With the assistance of manual plans, there is a 30% increase in success rate. (*ii*) For simple subtasks, the agent typically requires more extensive procedural execution compared to manual demonstrations (+2.1), which often represent the optimal pathway. This redundancy is exacerbated in complex tasks. Therefore, enhancing planning capabilities is essential for achieving efficient system with accurate success rates.

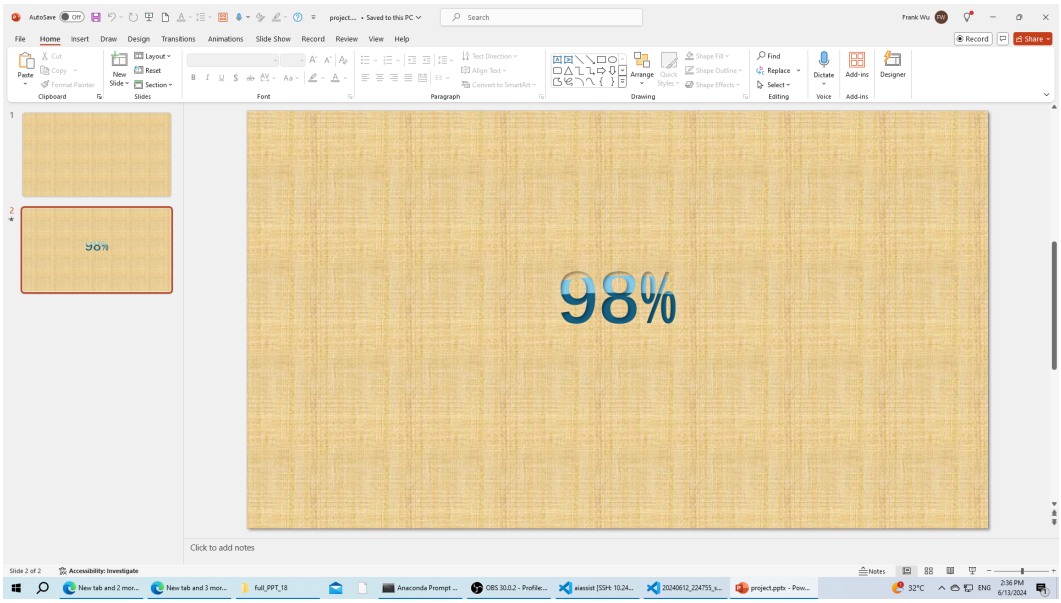
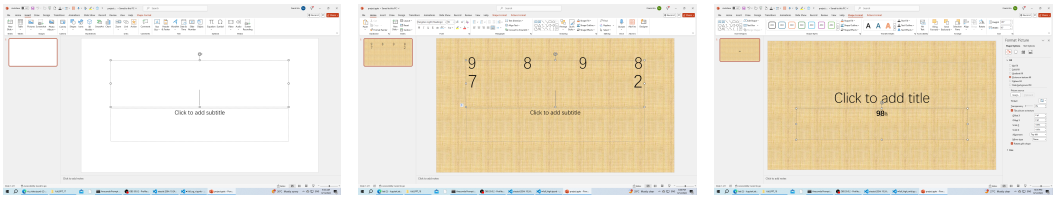


Figure 10: Final effect in Powerpoint files.



(a) GPT-4o **(b) GPT-4o w. GT High Plan** **(c) GPT-4o w. GT High+Mid. Plan**

Figure 11: Example of final outcome with our simple GPT-4o agent in simulated environment. Guided by the GT planning, both (b) and (c) successfully insert the textual background, while the (c) can accurately type '98%'.

4 Qualitative Examples

Data samples. In this section, we display the visual-preview data samples, which are mainly focused on visual creation or editing.

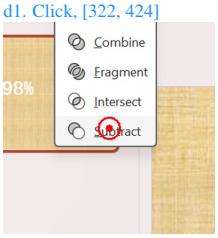
Visual preview			
Full task	**High-level Plans**	**Mid.-level Plans**	**Atomic Actions**
Visual query: How to create this effect in Powerpoint? **Textual query:** Create a slide that displays a large percentage figure of "98%" against a textured, beige background that appears to be fabric or canvas. The numerals are rendered in a bold, stylized font. The visual effect in this image is a wave-like effect. The blue percentage numerals appear to be rising out of the beige fabric-like background, creating a dynamic appearance. This gradient of wave creates a sense of depth and dimensionality, making the wave appear to have volume and curvature. The lighter blue at the top catches the light more, giving an illusion of the wave crest rising up, while the darker blue below suggests shadow and recession.	a. Format the background for the canvas b. Change the background texture to parchment. Add a text box, add 98%, increase the font size and bold effect c. Change the background texture to papyrus, increase the font size of 98%, change color to white, center it in the middle d. Add a rectangle, remove outline, change the texture to papyrus e. Send the rectangle to the back **f. Select the rectangle and the text. Merge shape and subtract, add buttom right shadow** g. Add shapes (e.g. Ovals) in between the two layers h. Duplicate the slide, place it nicely and add Morph transition effect	f1. Drag to select the rectangle and text '98%' f2. Click on Shape Format button f3. Click on Merge Shapes button **f4. Click on Subtract button** f5. Click on Presets button f6. Click on shadow with buttom right	d1. Click, [322, 424] 

Table 15: Video Creation (*i.e.,* animation) example with **Powerpoint.**

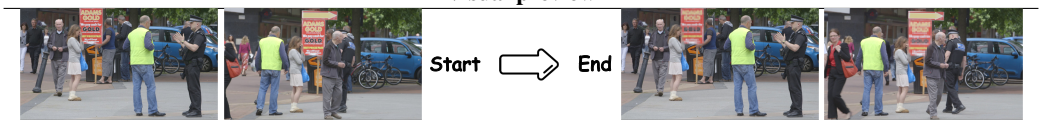
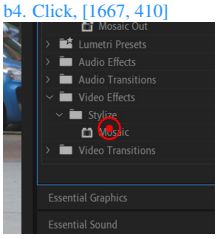
Visual preview			
			
Full task	**High-level Plans**	**Mid.-level Plans**	**Atomic Actions**
Visual query: How to transform from [start] to [end] in Premiere Pro? **Textual query:** Add a rectangle mosaic mask to the red billboard and track it.	**a.** Drag the timestamp to the beginning of the video **b. Add Mosaic effect on the top clip** **c.** Adjust the granularity of the Mosaic to 120 **d.** Add a rectangle mask to cover the bilboard and track it	b1. Click on Effects b2. Click on Search box in Effects panel b3. Key board Type Mosaic **b4. Click on 'Mosaic' effect** b5. Drag the Mosaic effect to the top clip.	b4. Click, [1667, 410] 

Table 16: Video Editing example with **Premiere Pro.**

Evaluation visualization.

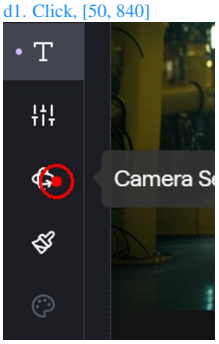
Visual preview			
Full task	**High-level Plans**	**Mid.-level Plans**	**Atomic Actions**
Visual query: How to create this effect in Runway? **Textual query:** Create a video about "A man in a dark green jacket stands in the center of a futuristic industrial setting with yellow machines and monitors, under bright overhead lights, creating a cinematic portrait effect" with the dolly zoom effect.	**a.** Open Text/Image to Video Tool **b.** Generate preview picture with text "A man in a dark green jacket stands in the center of a futuristic industrial setting with yellow machines and monitors, under bright overhead lights, creating a cinematic portrait effect. **c.** Select the third image as the image input **d. Adjust camera settings. Set Zoom to -3** **e.** Select the background in Motion Brush. Set its Proximity to 10 **f.** Select the subject in Motion Brush. Set its Proximity to 2 **g.** Generate the video	**d1. Click on Camera Settings.** d2. Click on the value of Zoom. d3. Key board Type -3	d1. Click, [50, 840] 

Table 17: Video Creation example with **Runway.**

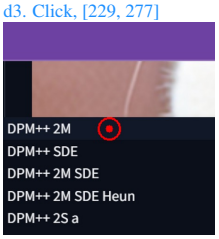
Visual preview			
Full task	**High-level Plans**	**Mid.-level Plans**	**Atomic Actions**
Visual query: How to transform from [start] to [end] in StableDiffusion-WebUI? **Textual query:** Replace the 512*512 photo of a cat to a 720*720 photo of dragon by DPM++ method.	**a.** Open img2img Tool and drag photo of cat into the file upload box **b.** Put "image of a dragon" into prompt box **c.** Put "cartoon" into negative prompt box **d. Set "Sampling method" to "DPM++ 2M Karras"** **e.** Set Width to 720 and Height to 720 **f.** Set Sampling steps to 25, Batch Size to 4 and CFG Scale to 4 **g.** Generate the image	d1. scroll down 7 d2. Click on options of Sampling method. **d3. Click on "DPM++ 2M".** d4. Click on options of Schedule type. d5. Click on Karras.	d3. Click, [229, 277] 

Table 18: Image Editing example with **StableDiffusion-WebUI.**

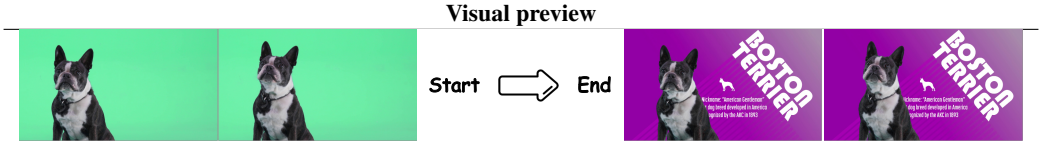
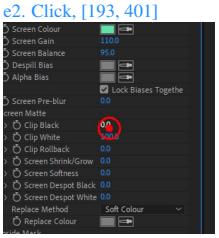
Visual preview			
			
Full task	**High-level Plans**	**Mid.-level Plans**	**Atomic Actions**
Visual query: How to transform from [start] to [end] in Adobe Effects? **Textual query:** Isolate the dog with Green Screen.	**a.** Select and apply Keylight effect to the BostonTerrier.mov layer **b.** Use the eyedropper tool to select the green background **c.** Adjust Keylight view mode to Screen Matte **d.** Modify Screen Gain and Screen Balance parameters **e. Adjust Clip Black and Clip White parameters in Screen Matte** **f.** Switch view mode back to Final Result and hide background layer	e1. Click on Expand icon of Screen Matte **e2. Click on Parameter of Clip Black 0.0** e3. Key board Type 10 e4. Click on Parameter of Clip White 100.0 e5. Key board Type 85	e2. Click, [193, 401] 

Table 19: Video Editing example with **Adobe Effects.**

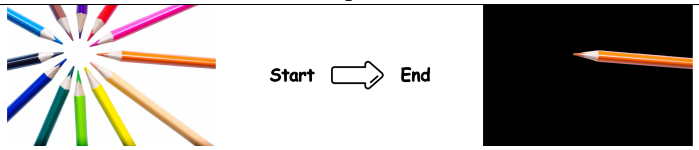
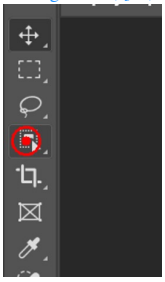
Visual preview			
			
Full task	**High-level Plans**	**Mid.-level Plans**	**Atomic Actions**
Visual query: How to transform from [start] to [end] in Photoshop? **Textual query:** Use quick selection tool to put the pencil in the black background.	**a. Use quick selection tool to select the pencil** **b.** Create a mask **c.** Create a solid black background layer **d.** Refine the mask. Set the smooth to 8, Feather to 7 px, Contrast to 72%, and Shift Edge to -3%;	**a1. RightClick on Quick Selection Tool.** a2. Click on Quick Selection Tool. a3. Drag the orange pencil from right to left. (Purpose: select the orange pencil)	b4. RightClick, [25, 271] 

Table 20: Image Editing example with **Photoshop.**

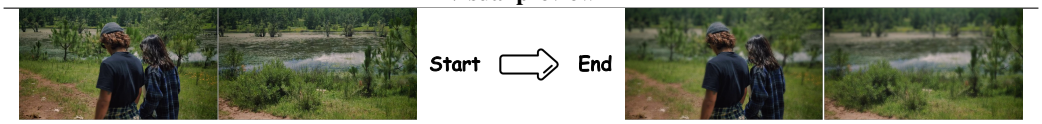
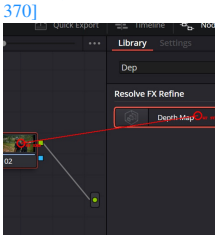
Visual preview			
			
Full task	**High-level Plans**	**Mid.-level Plans**	**Atomic Actions**
Visual query: How to transform from [start] to [end] in DaVinci? **Textual query:** Use Depth Map to blur the background.	**a. Add a serial node with depth map** **b.** Add a serial node with lens blur **c.** Connect nodes and inverse the depth map node **d.** Disable Depth Map Preview	a1. Click on Color panel. a2. Click on Effects. a3. Click on Search bar in Effects panel. a4. Key board Type Dep a5. RightClick on the video node in node editor. a6. Click on "Add Node > Add Serial". **a7. Drag Depth Map from Effects panel to video node 02. (Purpose: add Depth Map to the video node 02)**	a7. Drag, [2175, 305]→[1758, 370] 

Table 21: Video Editing example with **DaVinci.**

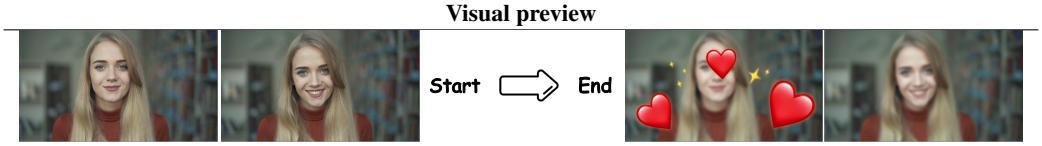
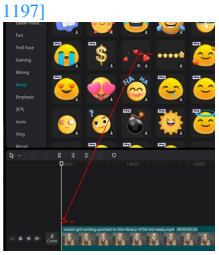
Visual preview			
			
Full task	**High-level Plans**	**Mid.-level Plans**	**Atomic Actions**
Visual query: How to transform from [start] to [end] in CapCut? **Textual query:** Add Stickers "Heart", Effects "Blur" and Filters "Glow" to the video.	**a. Add "Heart" Sticker to the video** **b.** Add "Blur" Effect to the video **c.** Add "Glow" Filter to the video	a1. Click on Click on Stickers Tool. **a2. Drag "heart" from Stickers Pool to video track. (Purpose: add "heart" to the video track)**	a2. Drag, [599, 464]→[265, 1197] 

Table 22: Video Editing example with **CapCut.**

References

[1] OBS Studio. Obs studio. https://obsproject.com/.

[2] Meta. Introducing meta llama 3: The most capable openly available llm to date, 2024. Accessed: 2024-04-18.

[3] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[4] OpenAI. Introducing chatgpt. OpenAI Blog, 09 2021.

[5] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. *arXiv preprint arXiv:2312.08914*, 2023.

[6] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[7] AI Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 2024.

[8] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[9] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[10] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v, 2023.

[11] Azure OCR. Azure ocr. https://azure.microsoft.com/en-us/products/ai-services/ai-vision.