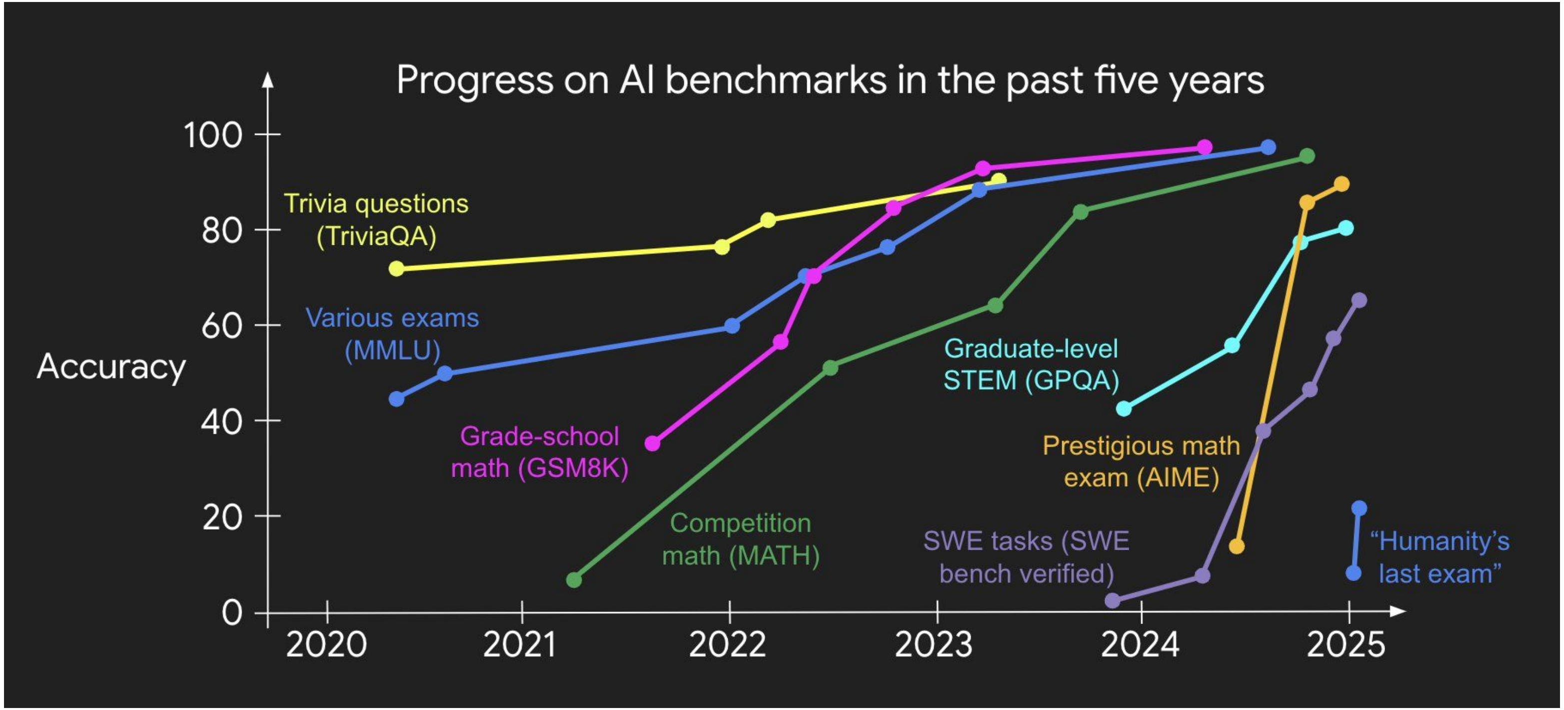
Multimodal Agent

From Perception to Action

Caiming Xiong

Salesforce AI Research
@caimingxiong

Intelligence grows rapidly, even surpassing humans.



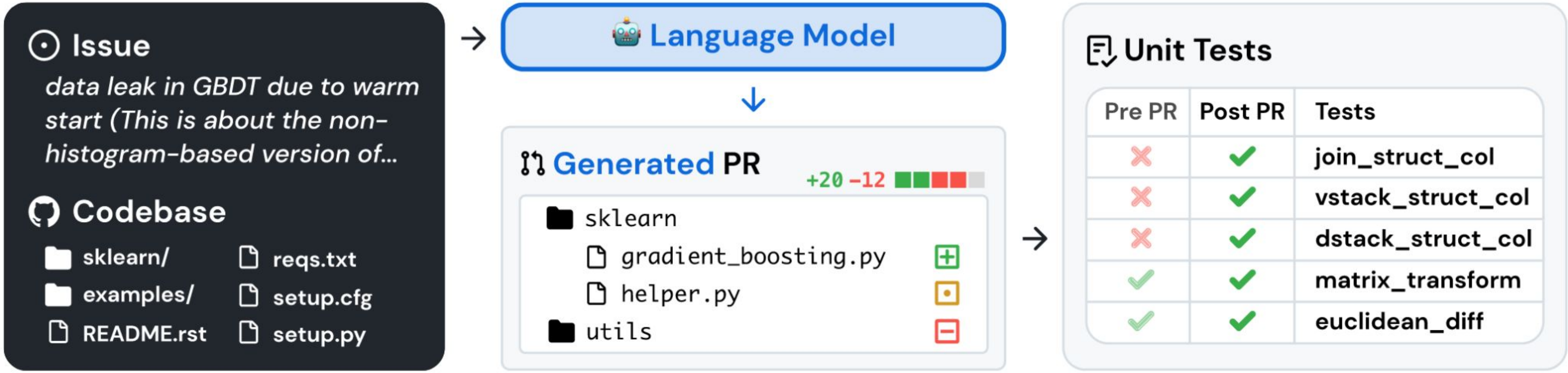
https://x.com/_jasonwei/status/1889096555254456397

Generated from Gemini

Multimodal Agents

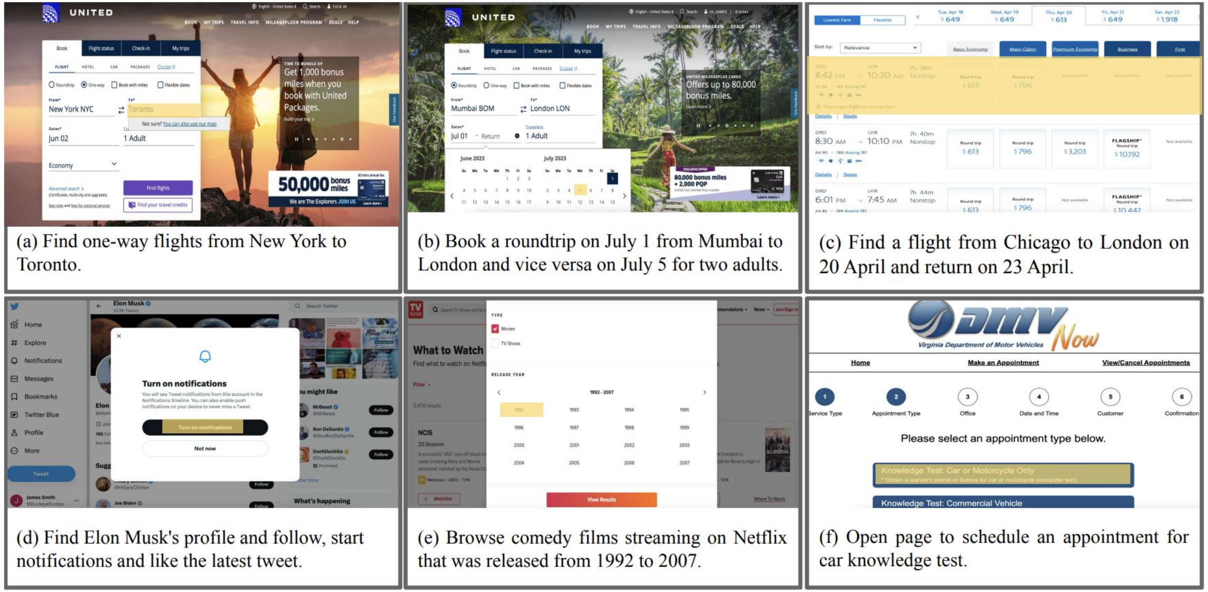
- Computer tasks often involve multiple apps and interfaces
- Powered by advancements in large vision-language-action models (VLA-Ms)
- Make digital interactions more accessible and vastly increase human productivity

Coding Agents



[SWE-bench: Can Language Models Resolve Real-World GitHub Issues?, (Jimenez et al., 2023)]
[MLE-bench: Evaluating Machine Learning Agents on Machine Learning Engineering, (Chan et al., 2024)]

Web Agents



(a) Find one-way flights from New York to Toronto.

(b) Book a roundtrip on July 1 from Mumbai to London and vice versa on July 5 for two adults.

(c) Find a flight from Chicago to London on 20 April and return on 23 April.

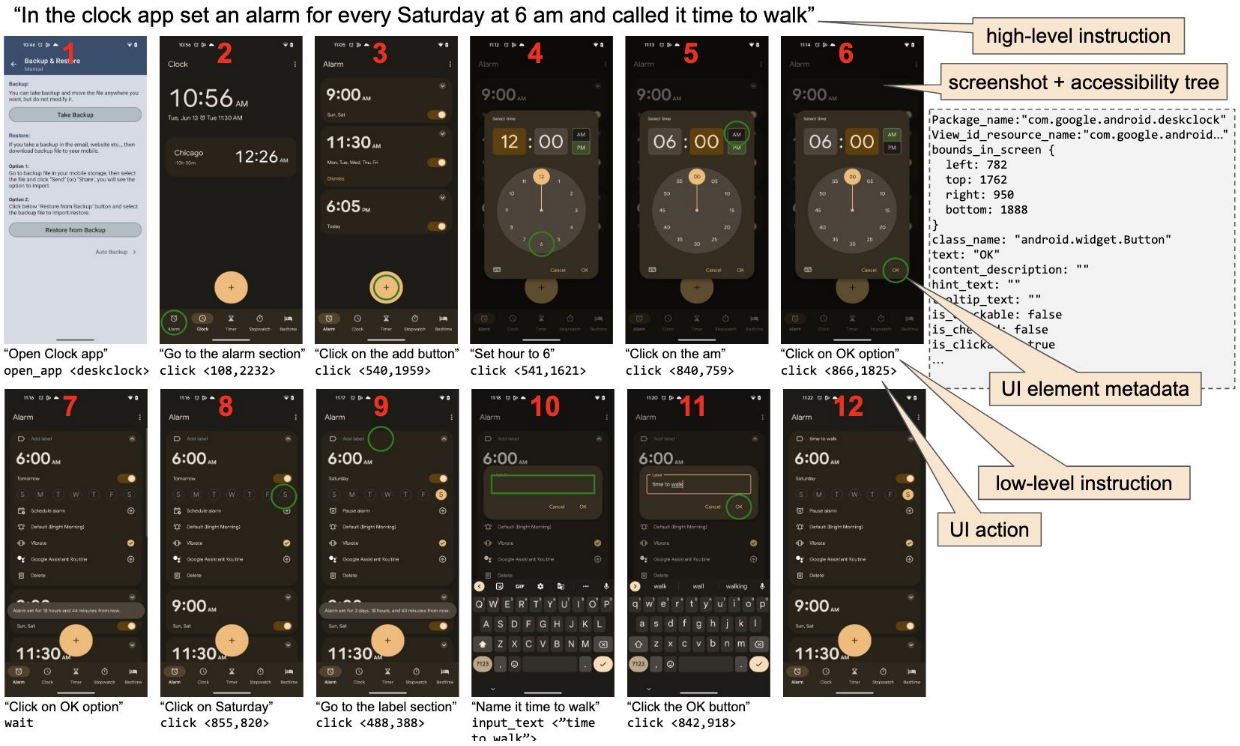
(d) Find Elon Musk's profile and follow, start notifications and like the latest tweet.

(e) Browse comedy films streaming on Netflix that was released from 1992 to 2007.

(f) Open page to schedule an appointment for car knowledge test.

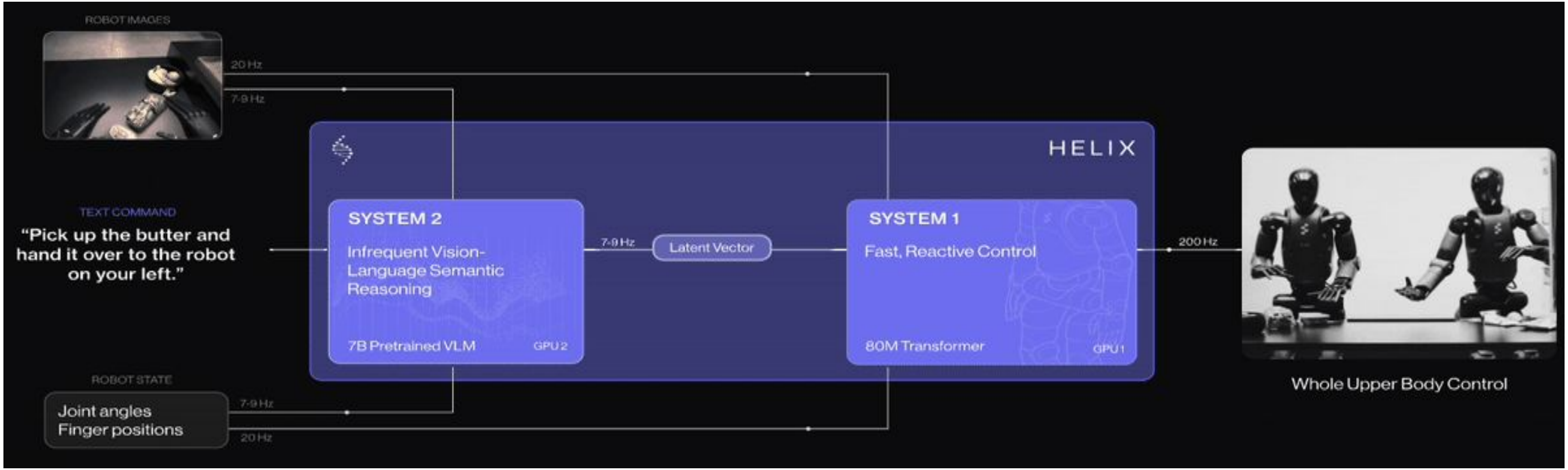
[World of Bits: An Open-Domain Platform for Web-Based Agents. (Shi et al., 2017)]
[Mind2Web: Towards a Generalist Agent for the Web. (Deng et al., 2023)]
[WebArena: A Realistic Web Environment for Building Autonomous Agents. (Zhou et al., 2023)]
[Browsergym: a Gym Environment for Web Task Automation (Drouin et al., 2024)]

Mobile Agents

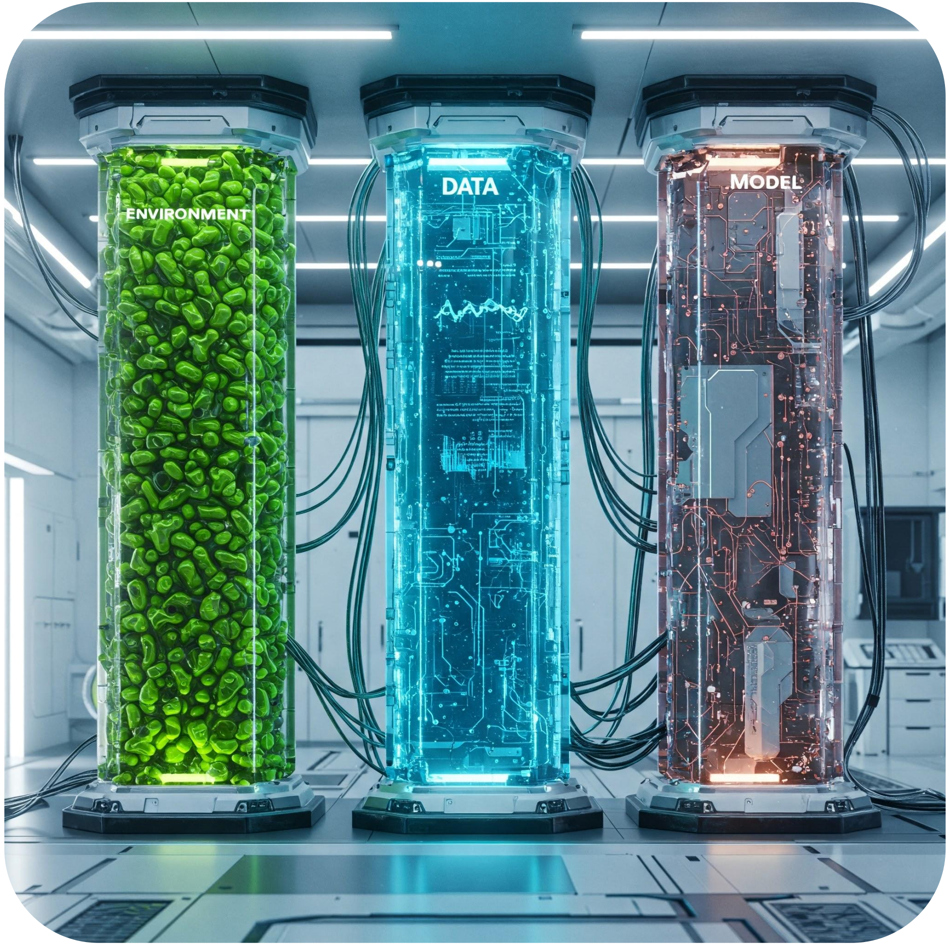


On the Effects of Data Scale on Computer Control Agents. (Li et al., 2024)

Physical Agents



https://www.figure.ai/news/helix



Generated from Gemini

Agenda

01 — **Environment/Benchmark: Should be reconfigurable and expandable**

02 — Data: Diverse modalities, large-scale, covering a wide range of tasks

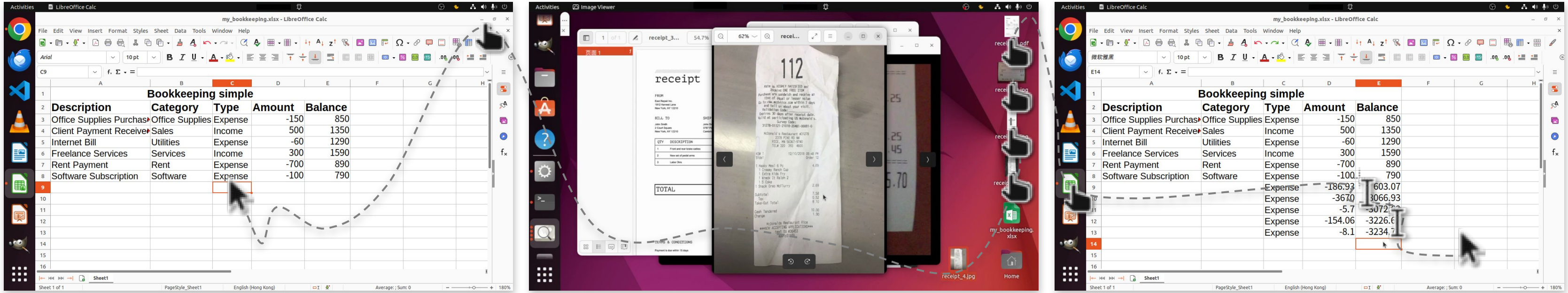
03 — Model/System: Unified vision-language-reasoning-action model, and long-context inference.

Computer Use

Computer tasks often involve multiple apps and interfaces.

salesforce

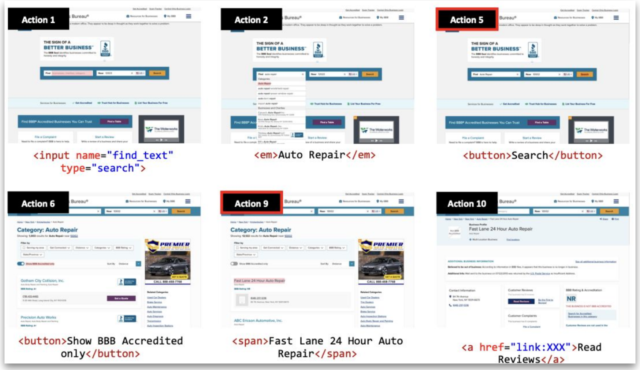
Task instruction 1: Update the bookkeeping sheet with my recent transactions over the past few days in the provided folder.



Current Benchmarks

no real, scalable interactive environments

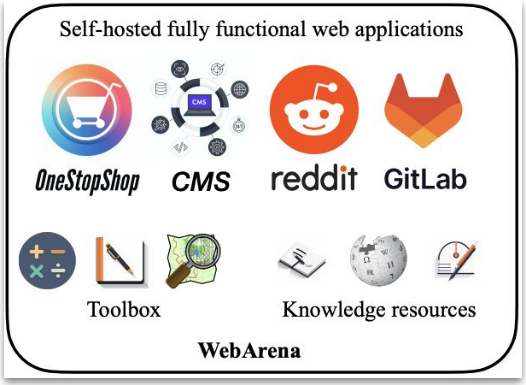
Mind2Web



Only demos *without* executable environment

- No execution based evaluation
- Cannot support interactive learning & real-world exploration

WebArena



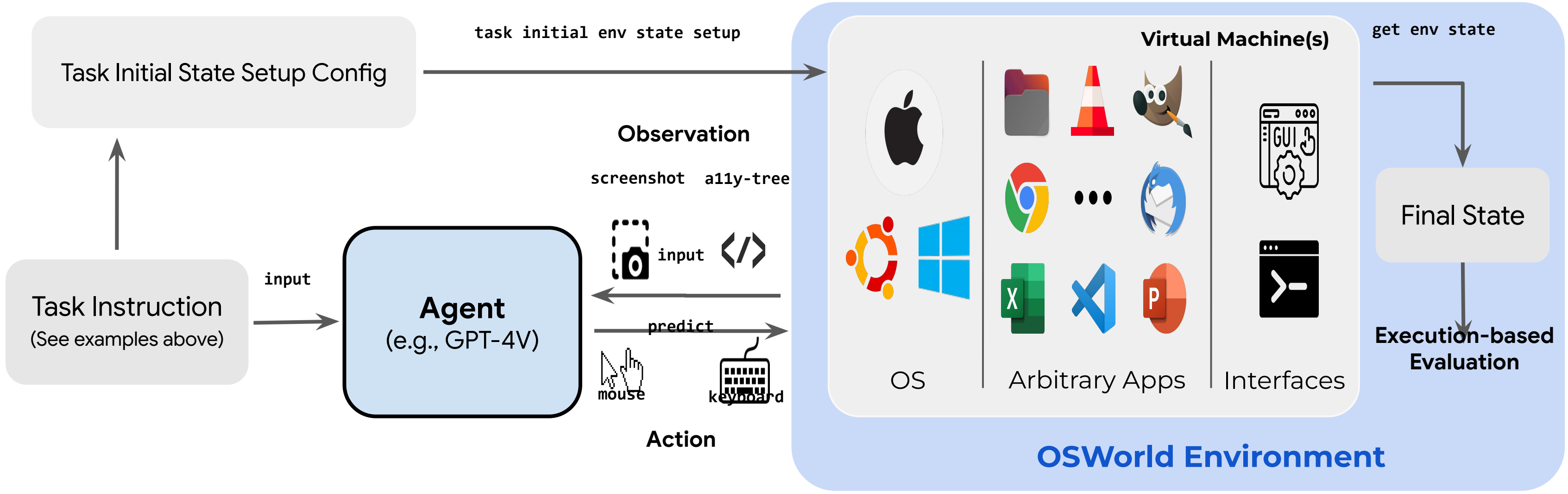
Environments limited to specific apps or domains

- Simplify agent's observation and action spaces
- Limit task scope, cannot support the evaluation of complex, real-world computer tasks

Mind2Web: Towards a Generalist Agent for the Web
WebArena: A Realistic Web Environment for Building Autonomous Agents

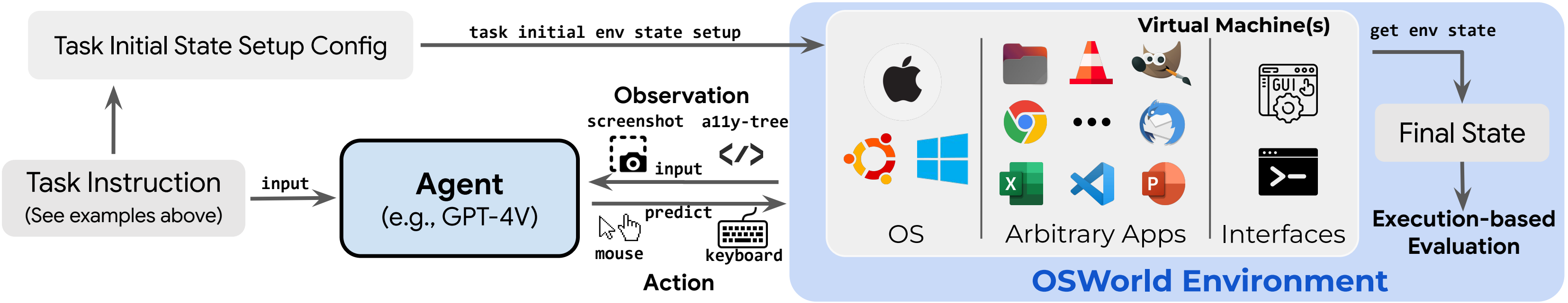
OSWorld

the first scalable, real computer environment



[OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments, (Xie et al., 2024)]

Agent Task Config



Given a computer task instruction:

- "Update the bookkeeping sheet with my recent transactions over the past few days in the provided folder."

Task Instruction
(See examples above)

Agent Task Config

Each computer task in OSWorld has a task initial state setup and evaluation config file.

Task Initial State Setup Config

Task Instruction
(See examples above)

Task Config

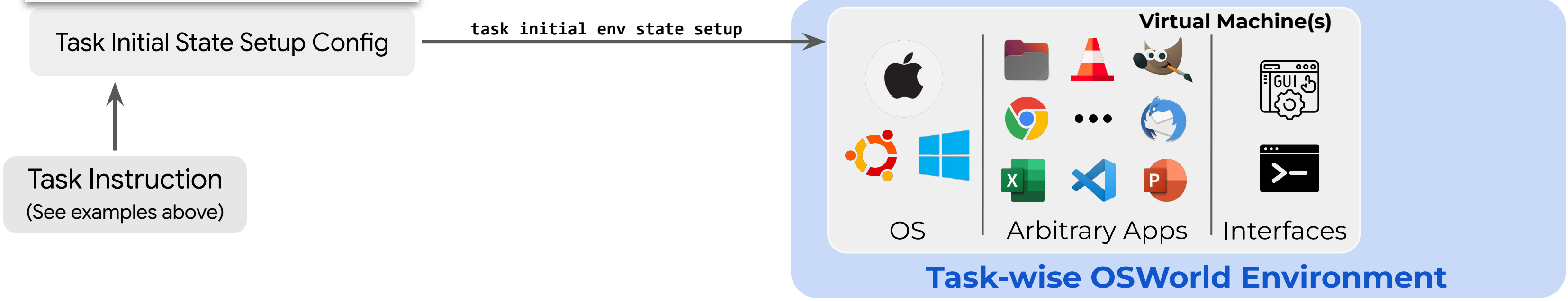
```
{ "instruction": "Please update my bookkeeping sheet with the
recent transactions from the provided folder, detailing my expenses
over the past few days.",
  "config": [{"type": "download",
      "parameters": {"files": [
{"path": "/home/user/Desktop/my_bookkeeping.xlsx",
 "url": "https://drive.google.com/uc?id=xxxx"},
{"path": "/home/user/Desktop/receipt_0.jpeg",
 "url": "https://drive.google.com/uc?id=xxxx"},…]}},
    {"type": "open",
      "parameters": {  "path":
"/home/user/Desktop/my_bookkeeping.xlsx"}}],
   "evaluator": {"postconfig": [{"type": "activate_window",
      "parameters": {"window_name": "my_bookkeeping.xlsx -
LibreOffice Calc",... ],
    "result": {"type": "vm_file",
      "path": "/home/user/Desktop/my_bookkeeping.xlsx",
      "dest": "my_bookkeeping.xlsx"},
    "expected": {"type": "cloud_file",
      "path": "https://drive.google.com/uc?id=xxx",
      "dest": "my_bookkeeping_gold.xlsx" },
    "func": "compare_table",
    "options": {
      "rules": [{
          "type": "sheet_fuzzy",
          "sheet_idx0": "RNSheet1",
          "sheet_idx1": "ENSheet1",
          "rules": [ {"range": [ "A1:A8",... }]}]
}
```

Agent Task Config

The task initial state setup config is used to create a virtual machine instance, and initializes intermediate state for each computer task.

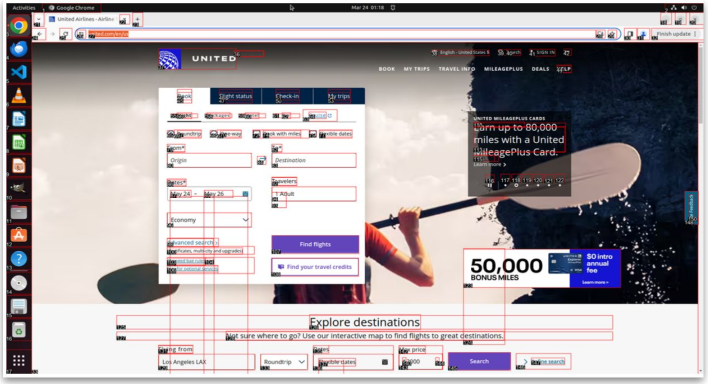
Task Config


```
{ "instruction": "Please update my bookkeeping sheet with the
recent transactions from the provided folder, detailing my
expenses over the past few days.",
  "config": [{"type": "download",
      "parameters": {"files": [
{"path": "/home/user/Desktop/my_bookkeeping.xlsx",
 "url": "https://drive.google.com/uc?id=xxxx"},
{"path": "/home/user/Desktop/receipt_0.jpeg",
 "url": "https://drive.google.com/uc?id=xxxx"},…]}},
    {"type": "open",
      "parameters": {  "path":
"/home/user/Desktop/my_bookkeeping.xlsx"}}],
   "evaluator": {"postconfig": [{"type": "activate_window",
      "parameters": {"window_name": "my_bookkeeping.xlsx -
LibreOffice Calc",... ],
    "result": {"type": "vm_file",
      "path": "/home/user/Desktop/my_bookkeeping.xlsx",
      "dest": "my_bookkeeping.xlsx"},
    "expected": {"type": "cloud_file",
      "path": "https://drive.google.com/uc?id=xxx",
      "dest": "my_bookkeeping_gold.xlsx" },
    "func": "compare_table",
    "options": {
      "rules": [{
          "type": "sheet_fuzzy",
          "sheet_idx0": "RNSheet1",
          "sheet_idx1": "ENSheet1",
          "rules": [ {"range": [ "A1:A8",... }]}]
}
```

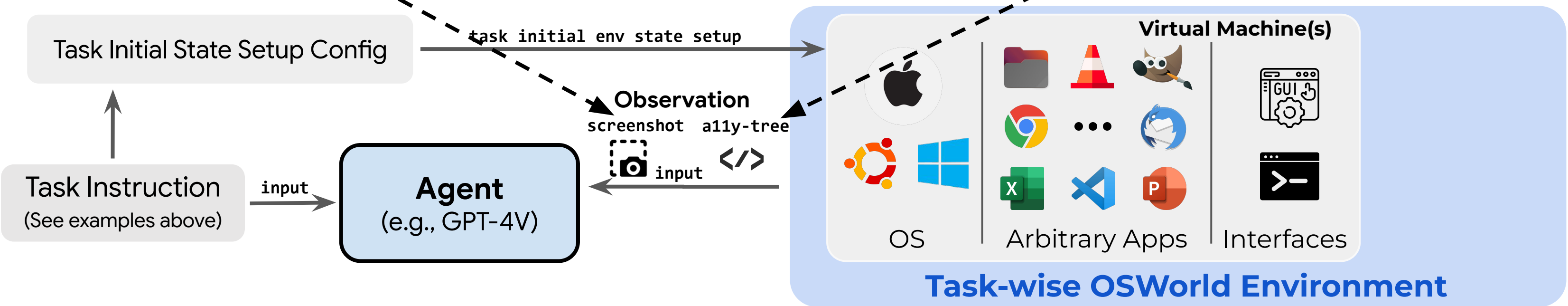
Agent Observation

Agent can receive natural language instruction, screenshots, the a11y tree, and customized streams such as terminal outputs.



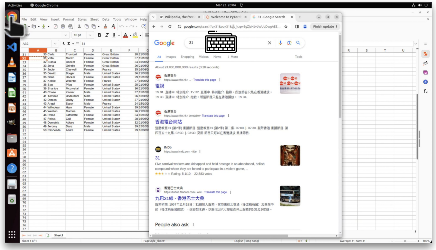
Set-of-Marks

Accessibility tree

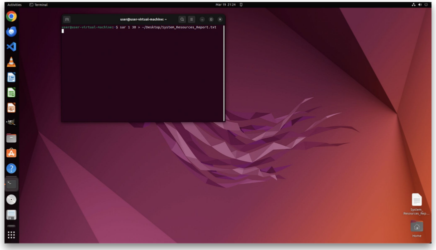


Agent Action Space

After receives the observations at each step, the agent generates executable actions



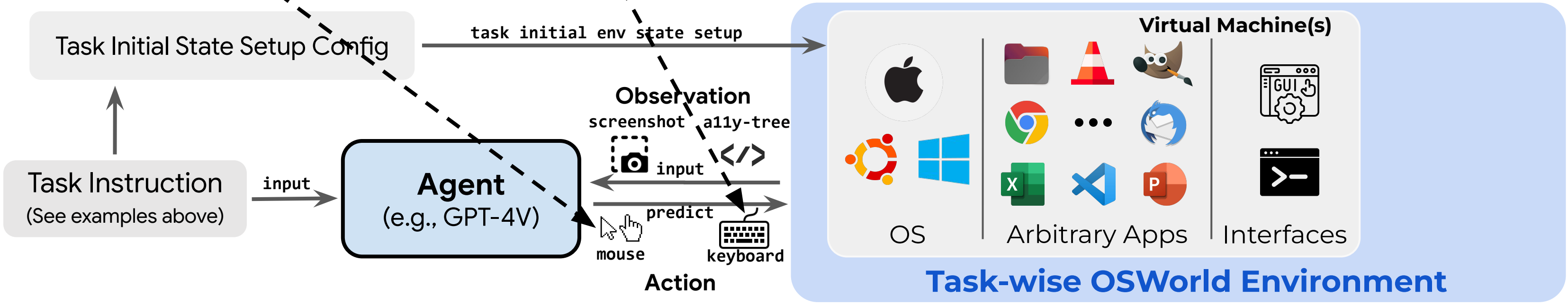
pyautogui.click(chrome_x, chrome_y) …



pyautogui.typewrite('sar 1 30 > …', interval=0.5)

Function	Description
moveTo(x, y)	Moves the mouse to the specified coordinates.
click(x, y)	Clicks at the specified coordinates.
write('text')	Types the specified text at the current cursor location.
press('enter')	Presses the Enter key.
hotkey('ctrl', 'c')	Performs the Ctrl+C hotkey combination (copy).
scroll(200)	Scrolls up by 200 units.
scroll(-200)	Scrolls down by 200 units.
dragTo(x, y)	Drags the mouse to the specified coordinates.
keyDown('shift')	Holds down the Shift key.
keyUp('shift')	Releases the Shift key.
WAIT	Agent decides it should wait.
FAIL	Agent decides the task is infeasible.
DONE	Agent decides the task is finished.

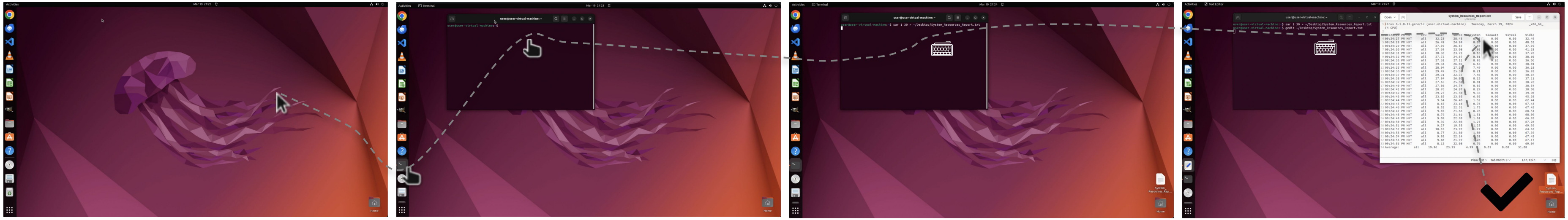
Some examples of the mouse and keyboard actions



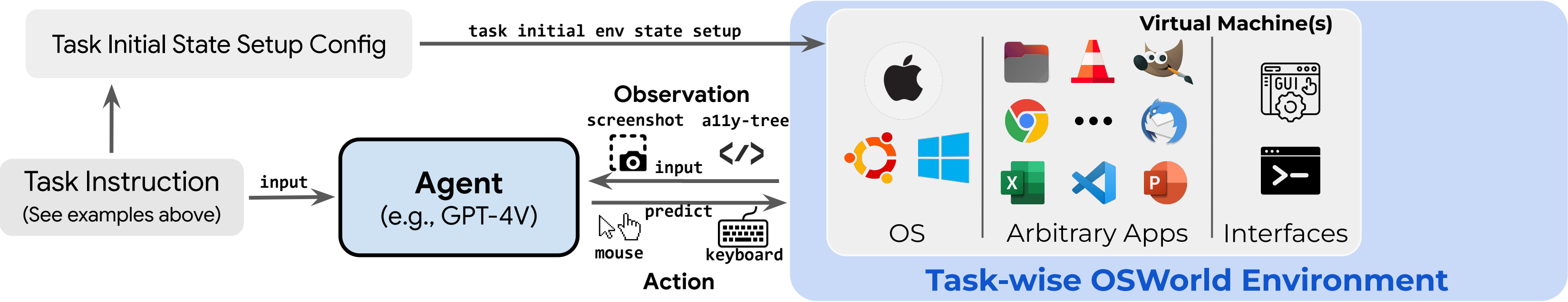
Agent Interaction Loop

The interaction loop between the agent and the environment repeats until an action that marks termination.

Task Instruction: monitor the system CPU for 30s and output the results



pyautogui.click(terminal_x, terminal_y) pyautogui.click(focus_x, focus_y) pyautogui.typewrite('sar 1 30 > …', interval=0.5) Done

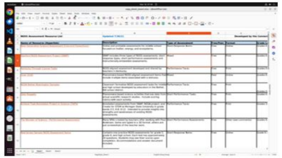


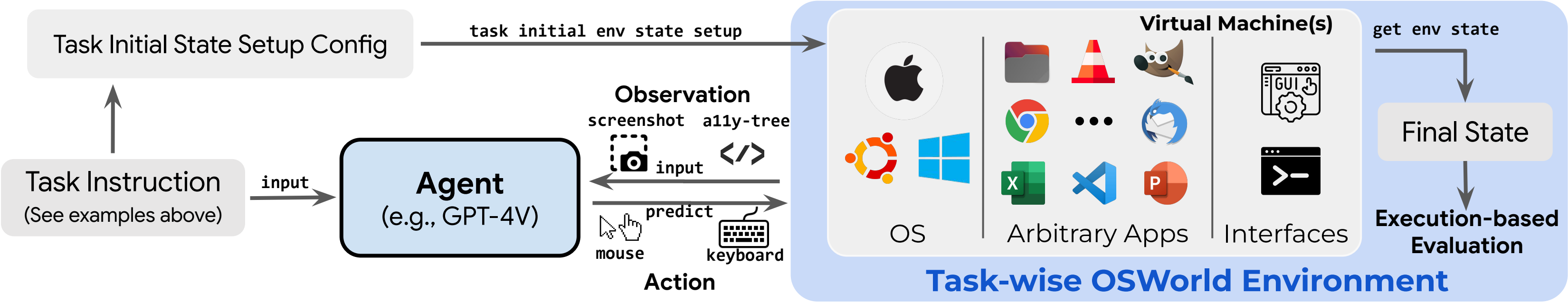
Agent Task Evaluation

In OSWorld, we implement an execution-based reward function

Initial State	Task Instruction	Evaluation Script (Simplified)

	*Can you help me clean up my computer by getting rid of all the tracking things that Amazon might have saved?*	`cookie_data = get_cookie_data(env)`
		`rule = {"type":"domains",`
		`"domains":[".amazon.com"]}`
		`is_cookie_deleted(cookie_data, rule)`

	*Rename "Sheet 1" to "LARS Resources". Then make a copy of it. Place the copy before "Sheet 2" and rename it by appending a suffix "(Backup)", ...*	`result = get_file(env)`
		`expected = get_file(cloud)`
		`rules = [{"type":"sheet_name"},`
		`{"type":"sheet_data",`
		`"sheet_idx0":0,`
		`"sheet_idx1":1}...]`
		`compare_table(result, expected, rules)`



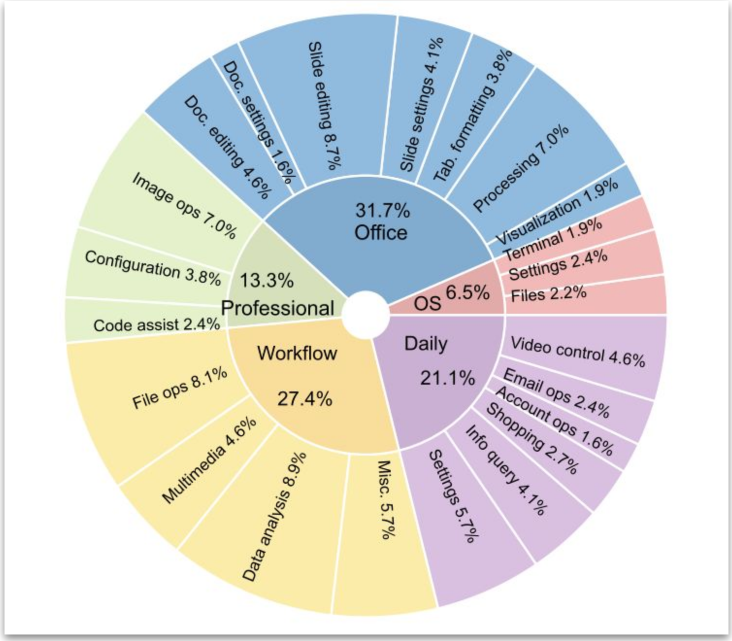
OSWorld benchmark dataset

369 real-world computer tasks that involve real web and desktop apps in open domains, OS file I/O, and multi-app workflows. Each task example is annotated with

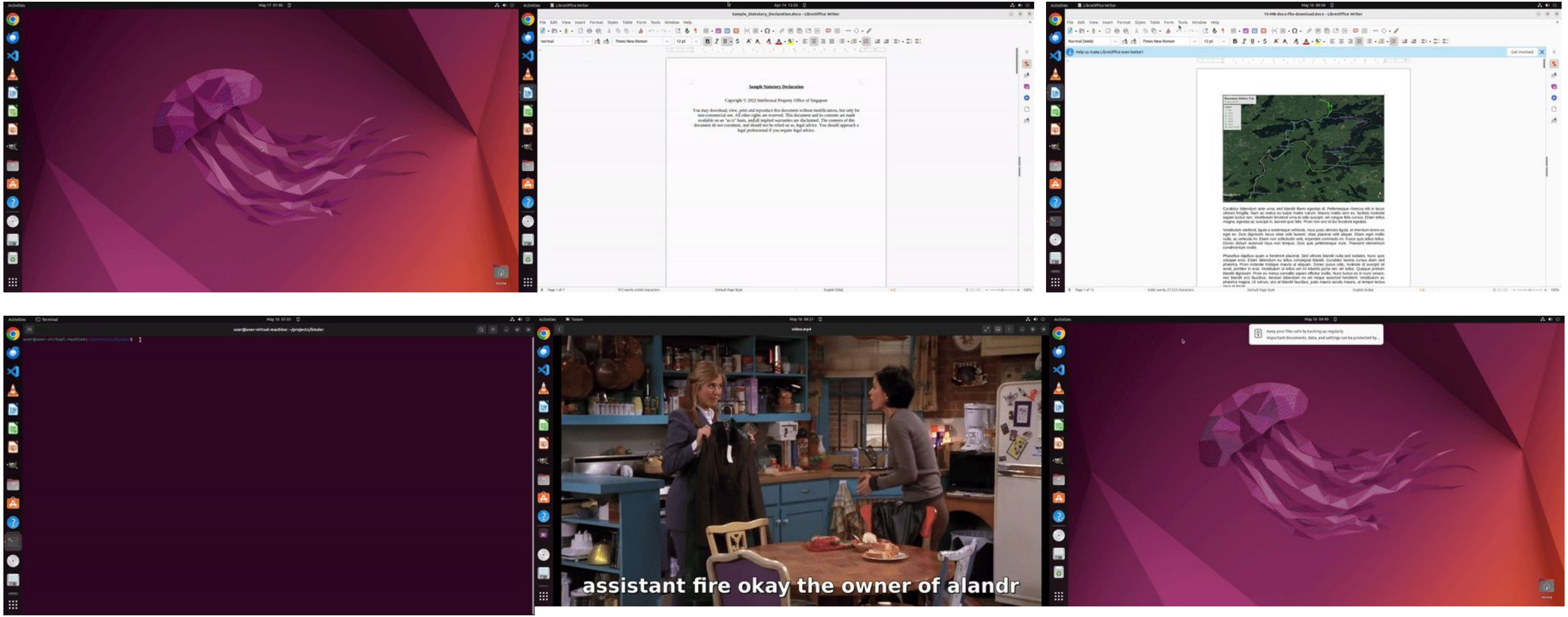
- A real-world task instruction from real users
- An initial state setup config to simulate human work in progress
- A custom execution-based evaluation script

Table 3: Key statistics in OSWORLD. The "Supp. tasks" refers to the Windows-based tasks, that could only be used after activation due to copyright restrictions.

Statistic	Number
Total tasks (Ubuntu)	369 (100%)
- Multi-App Workflow	101 (27.4%)
- Single-App	268 (72.6%)
- Integrated	84 (22.8%)
- Infeasible	30 (8.1%)
Supp. tasks (Windows)	43
Initial States	302
Eval. Scripts	134



OSWorld benchmark dataset



OSWorld benchmark dataset

	# Instances (# Templates)	Control. Exec. Env.?	Environment Scalability?	Multimodal Support?	Cross-App?	Intermediate Init. State?	# Exec.-based Eval. Func.
GAIA [35]	466	✗	-	✗	✗	✗	0
Mind2Web [9]	2350	✗	-	✓	✗	✓	0
WebLINX [33]	2337	✗	-	✓	✗	✓	0
PixelHelp [27]	187	✗	-	✓	✗	✗	0
MetaGUI [45]	1125	✗	-	✓	✗	✗	0
AitW [39]	30k	✗	-	✓	✗	✓	0
OmniAct [21]	9802	✗	-	✓	✗	✓	0
AgentBench [32]	1091	Multi-isolated	✗	✗	✗	✗	7
InterCode [54]	1350 (3)	Code	✗	✗	✗	✗	3
MiniWoB++ [30]	125	Web	✗	✓	✗	✗	125
WebShop [55]	12k (1)	Web	✗	✓	✗	✗	1
WebArena [63]	812 (241)	Web	✗	✓	✗	✗	5
VWebArena [22]	910 (314)	Web	✗	✓	✗	✗	6
WorkArena [10]	23k (29)	Web	✗	✓	✗	✓	7
WikiHow [58]	150 (16)	Mobile	✗	✓	✗	✗	16
AssistGUI [13]	100	✗	✗	✓	✗	✓	2
OSWorld	369	Computer	✓	✓	✓	✓	**134**

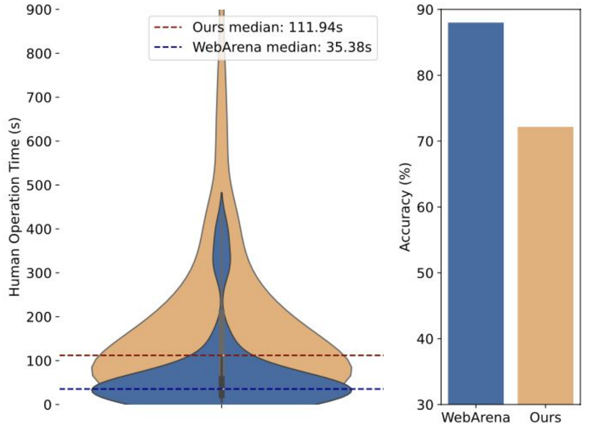


Figure 4: Human operation time and accuracy on OSWorld and WebArena.

OSWorld benchmark dataset

```
You are an agent which follow my instruction and perform desktop computer
↪  tasks as instructed.
You have good knowledge of computer and good internet connection and assume
↪  your code will run on a computer for controlling the mouse and keyboard.
For each step, you will get an observation of an image, which is the
↪  screenshot of the computer screen and you will predict the action of the
↪  computer based on the image.

You are required to use `pyautogui` to perform the action grounded to the
↪  observation, but DONOT use the `pyautogui.locateCenterOnScreen` function
↪  to locate the element you want to operate with since we have no image of
↪  the element you want to operate with. DONOT USE `pyautogui.screenshot()`
↪  to make screenshot.
Return one line or multiple lines of python code to perform the action each
↪  time, be time efficient. When predicting multiple lines of code, make
↪  some small sleep like `time.sleep(0.5);` interval so that the machine
↪  could take; Each time you need to predict a complete code, no variables
↪  or function can be shared from history
You need to to specify the coordinates of by yourself based on your
↪  observation of current observation, but you should be careful to ensure
↪  that the coordinates are correct.
You ONLY need to return the code inside a code block, like this:
```python
# your code here
```
Specially, it is also allowed to return the following special code:
When you think you have to wait for some time, return ```WAIT```;
When you think the task can not be done, return ```FAIL```, don't easily say
↪ ```FAIL```, try your best to do the task;
When you think the task is done, return ```DONE```.

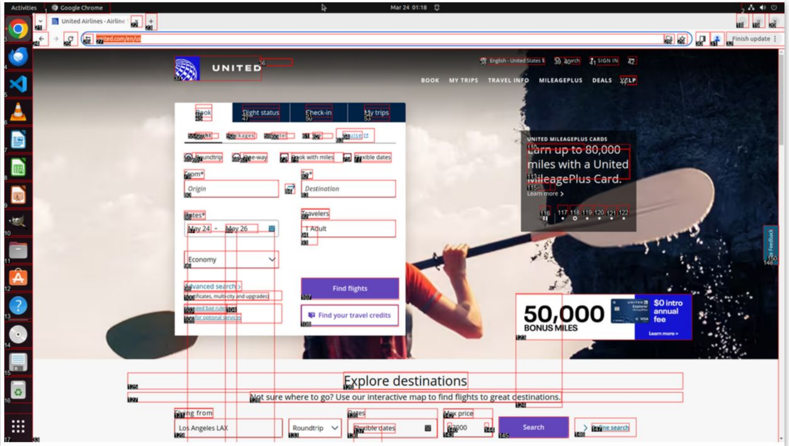
My computer's password is 'password', feel free to use it when you need sudo
↪ rights.
First give the current screenshot and previous things we did a short
↪ reflection, then RETURN ME THE CODE OR SPECIAL CODE I ASKED FOR. NEVER
↪ EVER RETURN ME ANYTHING ELSE.
```

- LLMs and VLMs from Mixtral and CogAgent (open-source), and GPT4, Gemini-pro, and Claude-3 (closed-source) as agents.
- Prompt details (see left - much more complex prompting pipelines)
- Temperature of 1.0 and top-p of 0.9
- Providing the most recent 3 observations and actions as history context for each step.

OSWorld benchmark dataset

Evaluation settings:

- Accessibility tree
- Screenshot
- Screenshot + accessibility tree
- Set-of-Marks




```
tag name text position (top-left x&y) size (w&h)
menu Google Chrome "" (99, 0) (162, 27)
menu System "" (1814, 0) (106, 27)
label Please download waiting software updates. "" (810, 73) (291, 17)
push-button Google Chrome "" (0, 33) (70, 64)
push-button Thunderbird Mail "" (0, 101) (70, 64)
push-button Visual Studio Code "" (0, 169) (70, 64)
push-button VLC media player "" (0, 237) (70, 64)
push-button LibreOffice Writer "" (0, 305) (70, 64)
push-button LibreOffice Calc "" (0, 373) (70, 64)
push-button LibreOffice Impress "" (0, 441) (70, 64)
push-button GNU Image Manipulation Program "" (0, 509) (70, 64)
push-button Files "" (0, 577) (70, 64)
push-button Ubuntu Software "" (0, 645) (70, 64)
push-button Help "" (0, 713) (70, 64)
push-button Ubuntu 22.04.3 LTS amd64 "" (0, 784) (70, 64)
push-button Floppy Disk "" (0, 852) (70, 64)
push-button Trash "" (0, 920) (70, 64)
toggle-button Show Applications "" (0, 1010) (70, 70)
label Home Home (1833, 1037) (40, 17)
push-button Minimise Minimise (1398, 51) (30, 30)
push-button Maximise Maximise (1438, 51) (30, 30)
push-button Close Close (1478, 51) (30, 30)
push-button Search tabs Search tabs (656, 46) (28, 41)
push-button Close Close (892, 52) (28, 28)
push-button New Tab New Tab (928, 46) (28, 41)
push-button Back Back (655, 92) (34, 34)
push-button Reload Reload (727, 92) (34, 34)
push-button View site information "" (775, 97) (24, 24)
entry Address and search bar recreation.gov (807, 97) (353, 24)
push-button Install Recreation.gov "" (1162, 97) (24, 24)
push-button Bookmark this tab "" (1194, 97) (24, 24)
push-button Side panel Side panel (1239, 92) (34, 34)
push-button You You (1275, 92) (34, 34)
push-button New Chrome available New Chrome available (1314, 92) (196, 34)
document-web Recreation.gov - Camping, Cabins, RVs, Permits, Passes & More "" (650, 133) (866, 922)
link Recreation.gov Homepage "" (675, 155) (241, 35)
```

OSWorld benchmark dataset

salesforce

Inputs	Model	Success Rate (↑)					
		OS	Office	Daily	Profess.	Workflow	Overall
A11y tree	Mixtral-8x7B	12.50%	1.01%	4.79%	6.12%	0.09%	2.98%
	Llama-3-70B	4.17%	1.87%	2.71%	0.00%	0.93%	1.61%
	GPT-3.5	4.17%	4.43%	2.71%	0.00%	1.62%	2.69%
	GPT-4	20.83%	3.58%	25.64%	26.53%	2.97%	**12.24%**
	Gemini-Pro	4.17%	1.71%	3.99%	4.08%	0.63%	2.37%
	Gemini-Pro-1.5	12.50%	2.56%	7.83%	4.08%	3.60%	4.81%
	Qwen-Plus	29.17%	3.58%	8.36%	10.20%	2.61%	6.87%
	GPT-4o	20.83%	6.99%	16.81%	16.33%	**7.56%**	11.36%
Screenshot	CogAgent	4.17%	0.85%	2.71%	0.00%	0.00%	1.11%
	GPT-4V	12.50%	1.86%	7.58%	4.08%	**6.04%**	5.26%
	Gemini-ProV	8.33%	3.58%	6.55%	16.33%	2.08%	**5.80%**
	Gemini-Pro-1.5	12.50%	6.99%	2.71%	6.12%	3.60%	5.40%
	Claude-3-Opus	4.17%	1.87%	2.71%	2.04%	2.61%	2.42%
	GPT-4o	8.33%	3.58%	6.07%	4.08%	5.58%	5.03%
Screenshot + A11y tree	CogAgent	4.17%	0.85%	2.71%	0.62%	0.09%	1.32%
	GPT-4V	16.66%	6.99%	24.50%	18.37%	4.64%	**12.17%**
	Gemini-ProV	4.17%	4.43%	6.55%	0.00%	1.52%	3.48%
	Gemini-Pro-1.5	12.50%	3.58%	7.83%	8.16%	1.52%	5.10%
	Claude-3-Opus	12.50%	3.57%	5.27%	8.16%	1.00%	4.41%
	GPT-4o	41.67%	6.16%	12.33%	14.29%	**7.46%**	11.21%
Set-of-Mark	CogAgent	4.17%	0.00%	2.71%	0.00%	0.53%	0.99%
	GPT-4V	8.33%	8.55%	22.84%	14.28%	**6.57%**	**11.77%**
	Gemini-ProV	4.17%	1.01%	1.42%	0.00%	0.63%	1.06%
	Gemini-Pro-1.5	16.67%	5.13%	12.96%	10.20%	3.60%	7.79%
	Claude-3-Opus	12.50%	2.72%	14.24%	6.12%	4.49%	6.72%
	GPT-4o	20.83%	3.58%	3.99%	2.04%	3.60%	4.59%
Human Performance		75.00%	71.79%	70.51%	73.47%	73.27%	72.36%

- LLMs and VLMs are still far from being digital agents on real computers.
- Agent performance fluctuations vs. consistent human performance across different types of computer tasks.
- A11y tree and SoM’s effectiveness varies by models.
- VLM agents with screenshot-only setting show lower performance, but it should be the ultimate configuration in the long run.

Result analysis of LLM/VLM agent baselines

Higher screenshot resolution typically leads to improved performance

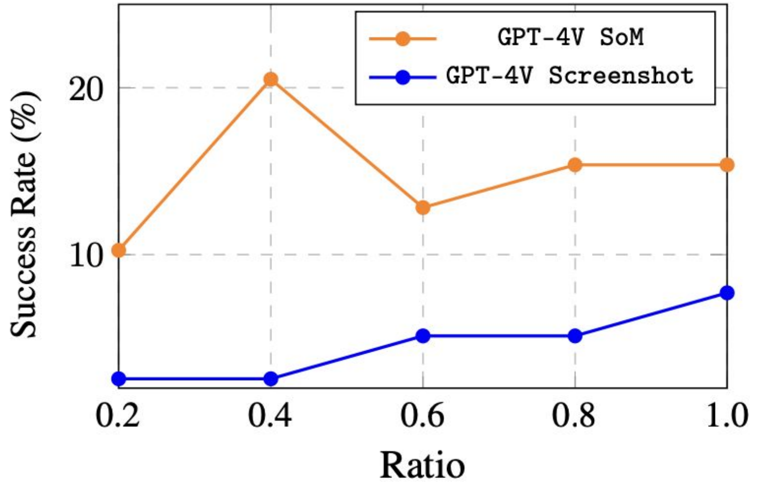


Figure 5: The effect of downsampling on the screenshot on performance with down-sampling ratios of 0.2, 0.4, 0.6 and 0.8 and run on a subset (10%) of examples.

Result analysis of LLM/VLM agent baselines

Longer text-based trajectory history context improves performance, unlike screenshot-only history, but poses efficiency challenges

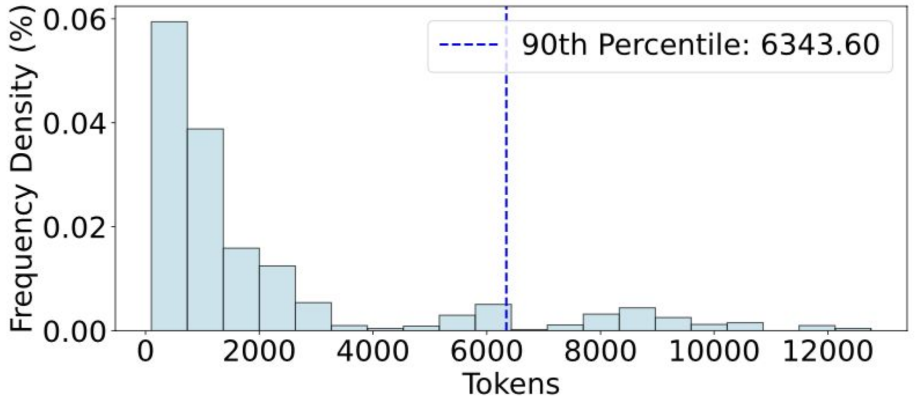


Figure 6: The length distribution of a11y tree as observation from sampled trajectories.

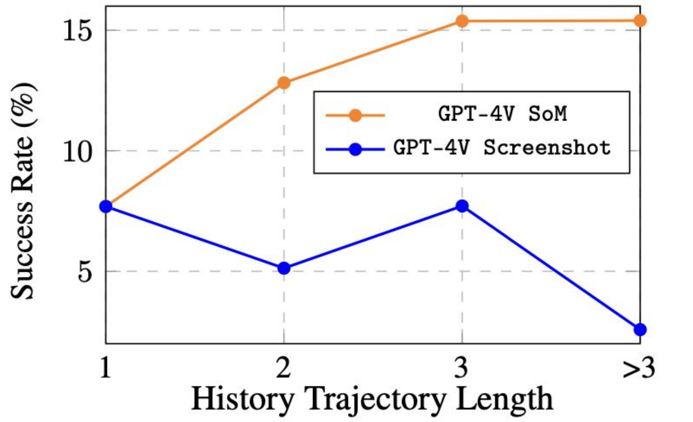


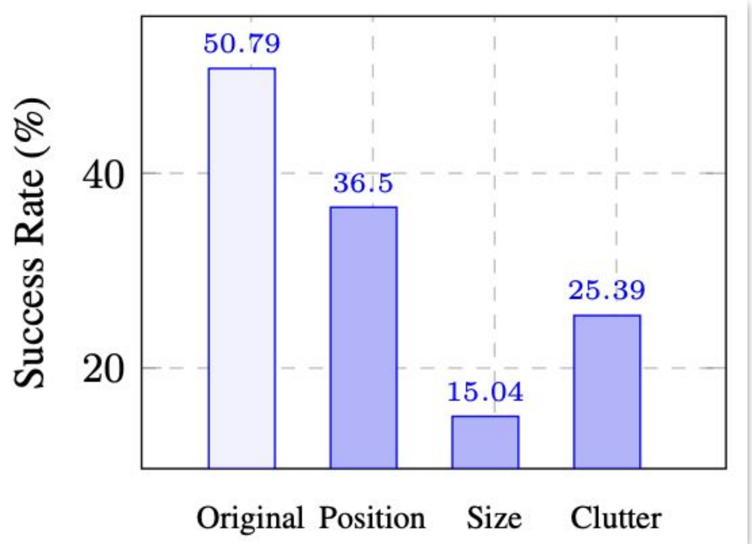
Figure 7: The effect of length of history on performance with the history encoding length of 1, 2, 3, and > 3 and run on a subset (10%) of examples.

Result analysis of LLM/VLM agent baselines

- The performance of VLM agents across different OS is in strong correlation.
- Current VLM agents are not robust to UI layout and noise

Table 7: Comparison of model performance and correlation across operating systems.

OS	SR (%)	Correlation Coefficient
Ubuntu	4.88	0.7
Windows	2.55	

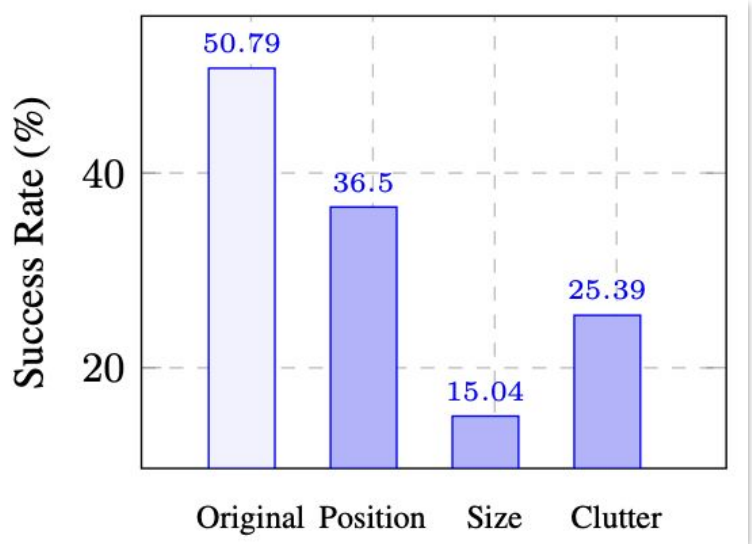


Result analysis of LLM/VLM agent baselines

- The performance of VLM agents across different OS is in strong correlation.
- Current VLM agents are not robust to UI layout and noise

Table 7: Comparison of model performance and correlation across operating systems.

OS	SR (%)	Correlation Coefficient
Ubuntu	4.88	0.7
Windows	2.55	



Recent progress

Introducing computer use, a new Claude 3.5 Sonnet, and Claude 3.5 Haiku

Oct 22, 2024 • 5 min read

Category	Claude 3.5 Sonnet (New) - 15 steps		Claude 3.5 Sonnet (New) - 50 steps		Human Success Rate [3]
	Success Rate	95% CI	Success Rate	95% CI	
OS	54.2%	[34.3, 74.1]%	41.7%	[22.0, 61.4]%	**75.00%**
Office	7.7%	[2.9, 12.5]%	17.9%	[11.0, 24.8]%	**71.79%**
Daily	16.7%	[8.4, 25.0]%	24.4%	[14.9, 33.9]%	**70.51%**
Professional	24.5%	[12.5, 36.5]%	40.8%	[27.0, 54.6]%	**73.47%**
Workflow	7.9%	[2.6, 13.2]%	10.9%	[4.9, 17.0]%	**73.27%**
Overall	14.9%	[11.3, 18.5]%	22%	[17.8, 26.2]%	**72.36%**

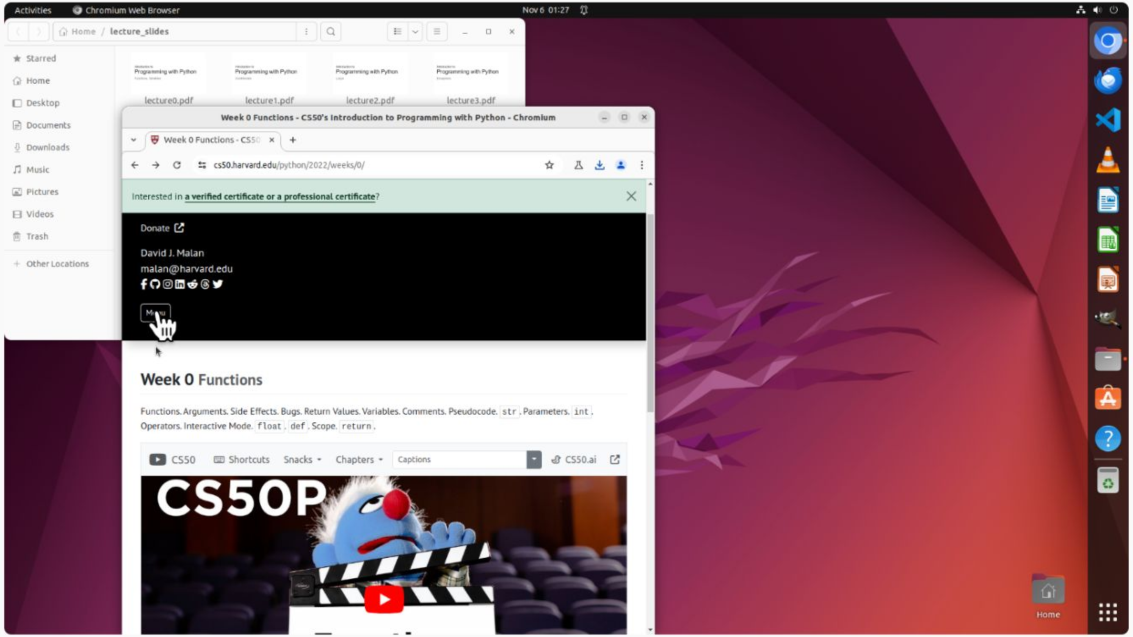
Anthropic computer use agent results on OSWorld

https://www.anthropic.com/news/3-5-models-and-computer-use

Recent progress

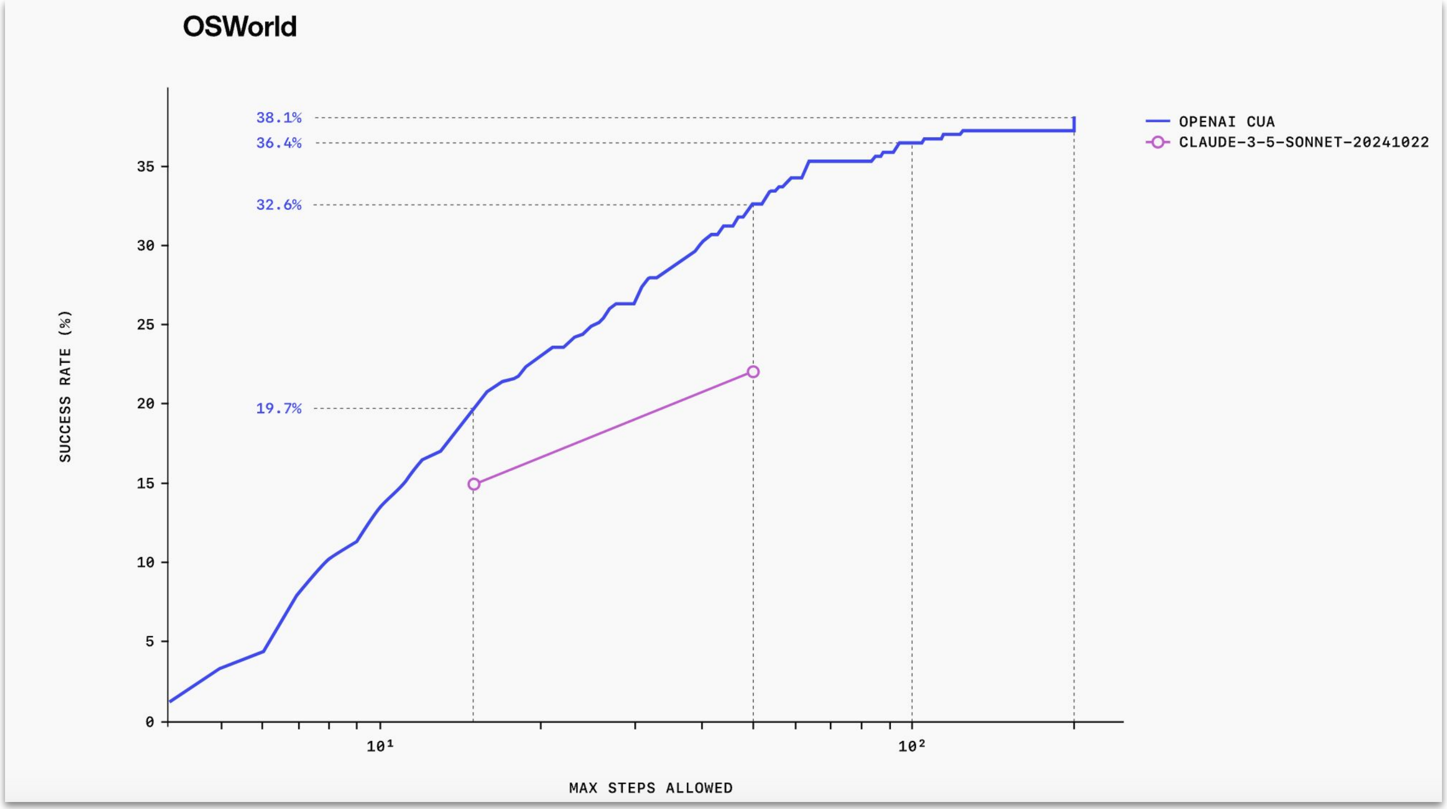
Download lectures | Combine pdfs | Compress image | Calculate price | Export images

186 Click
187 New screenshot
188 Click
189 New screenshot
190 Accessing menu for next week's content
191 Click
192 New screenshot
193 Click
194 New screenshot
195 Navigating to "Regular Expressions" page
196 Click
197 New screenshot
198 Scrolling for Week 7 slides
199 Scroll

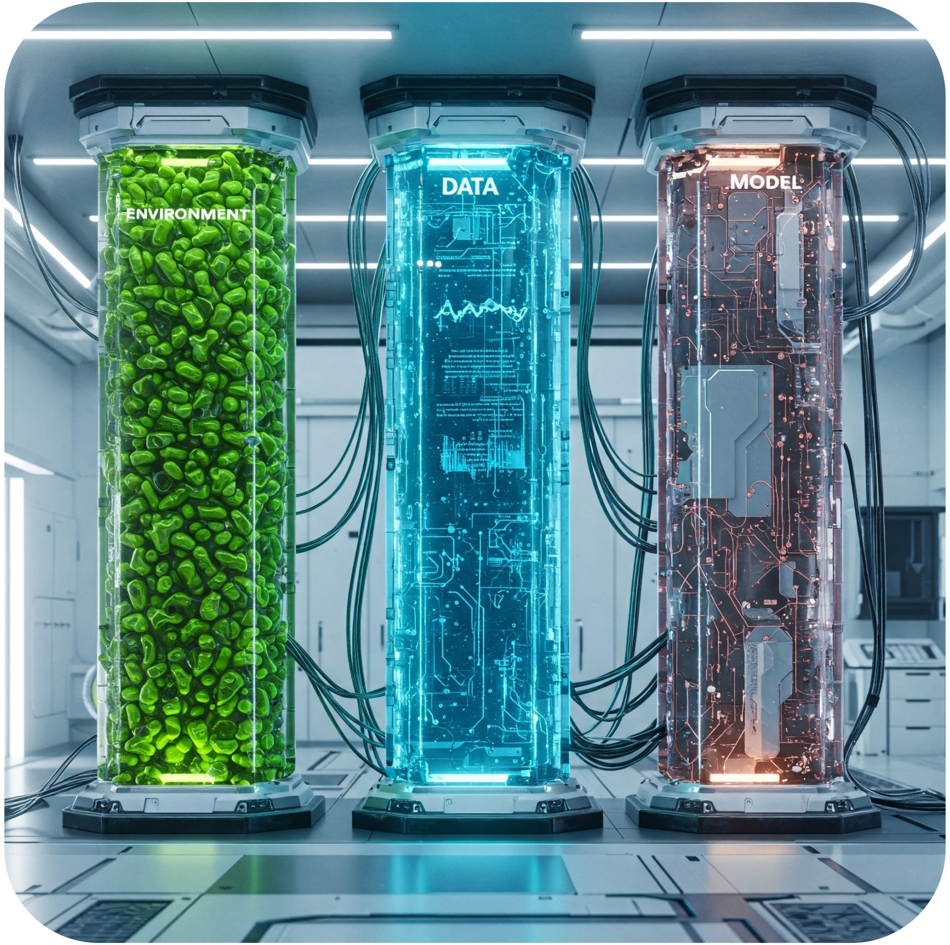


https://openai.com/index/computer-using-agent/

Recent progress



https://openai.com/index/computer-using-agent/



Generated from Gemini

Agenda

01 —— Environment/Benchmark: Should be reconfigurable and expandable

02 —— **Data: Diverse modalities, large-scale trajectory data, covering a wide range of tasks**

03 —— Model/System: Unified vision-language-reasoning-action model, and long-context inference.

Data Challenges for Agent Training

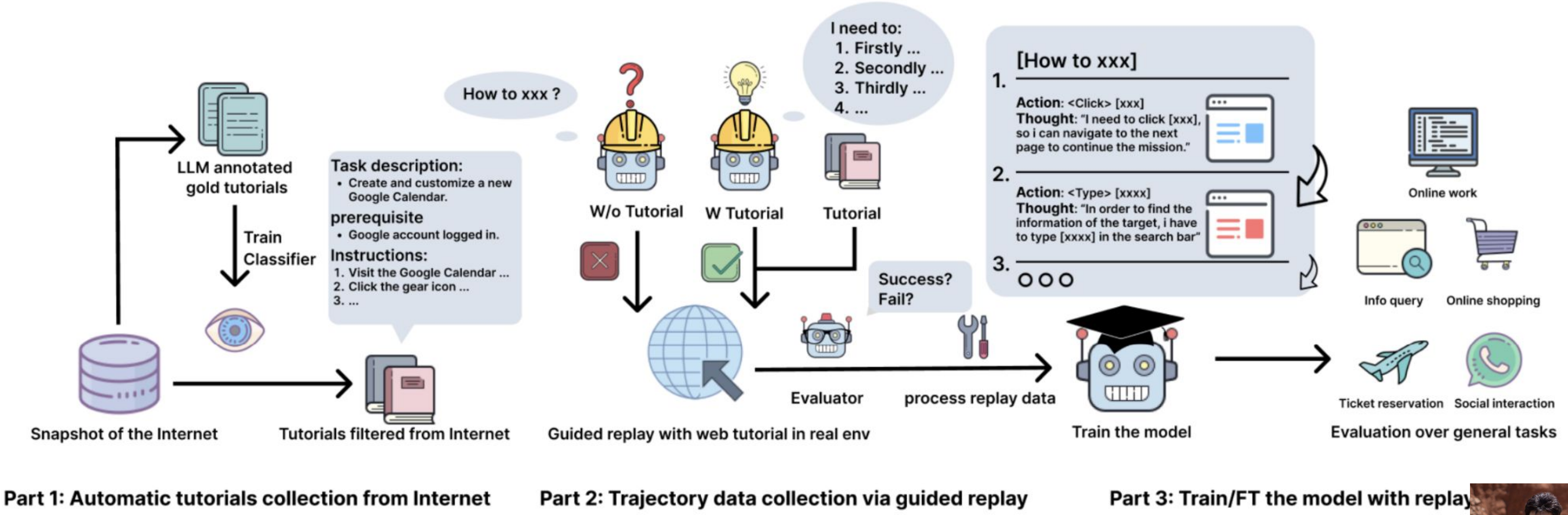
- Agent models require expensive human annotation to collect agent trajectory data.
- This contrasts with LLMs, which leverage existing text corpora.
- Human annotation is time-consuming, costly, and limits scalability.
- The cost and complexity of human annotation make it difficult to collect diverse and large-scale agent trajectory data.

No free large-scale trajectory corpus to crawl. Human annotation is so expensive!

Why don't we let the model to synthesize?

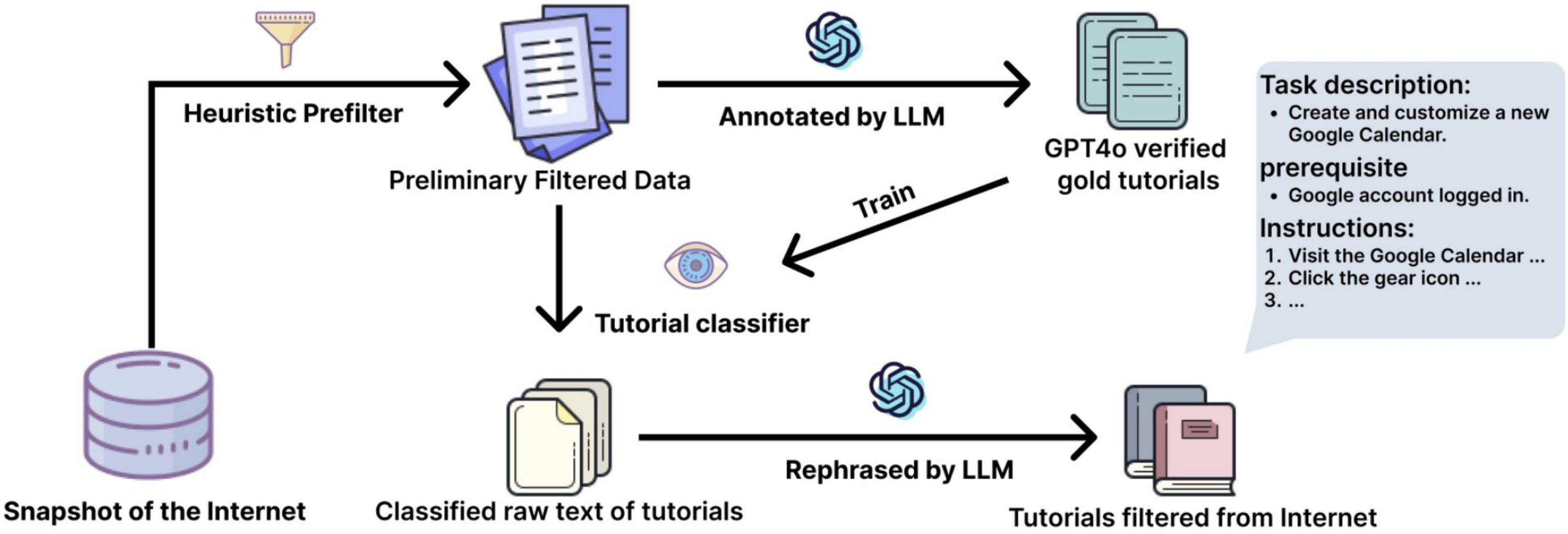
Agenttrek: agent trajectory synthesis via guiding replay with web tutorials

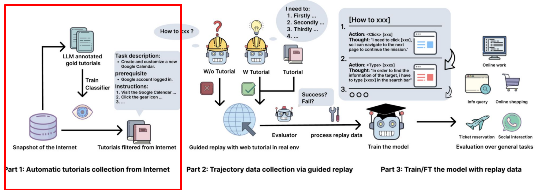
The internet contains a vast collection of tutorial-like text that provides step-by-step guidance on performing various tasks, particularly in GUI-based environments.



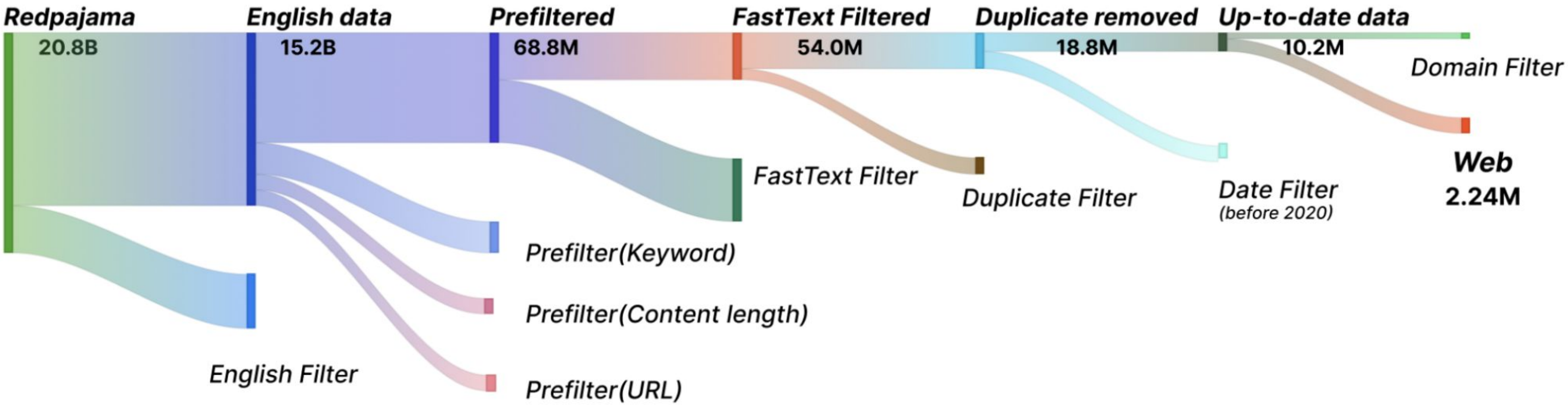
Agenttrek: agent trajectory synthesis via guiding replay with web tutorials. (Xu et.al. 2024)

Automatic Tutorial Collection From Internet





AgentTrek Tutorial Source Data Flow



User Prompt for Classifying GUI Tutorials

salesforce

System Prompt

```
You are an assistant that classifies content based on specific
criteria.  Your task is to evaluate whether a given piece of content
serves as a tutorial specifically related to graphical user
interfaces (GUI), such as for web applications, desktop applications,
or operating systems.
```

Classification Criteria

```
The content qualifies as a GUI-related tutorial if it meets the
following conditions:
1.  It includes a task description outlining what needs to be
achieved.
2.  It provides clear step-by-step instructions for interacting with
a GUI, such as:
    - Step 1:  Open the application
    - Step 2:  Navigate to the settings menu

Given the URL and context, determine if the content is a GUI-related
tutorial or not.  Output '1' if it is a GUI-related tutorial and '0'
if it is not.  Provide only the number as the output.
```

User Prompt

```
- URL: {url}
- Context:  {context}
```

Tag & Paraphrase

User Prompt

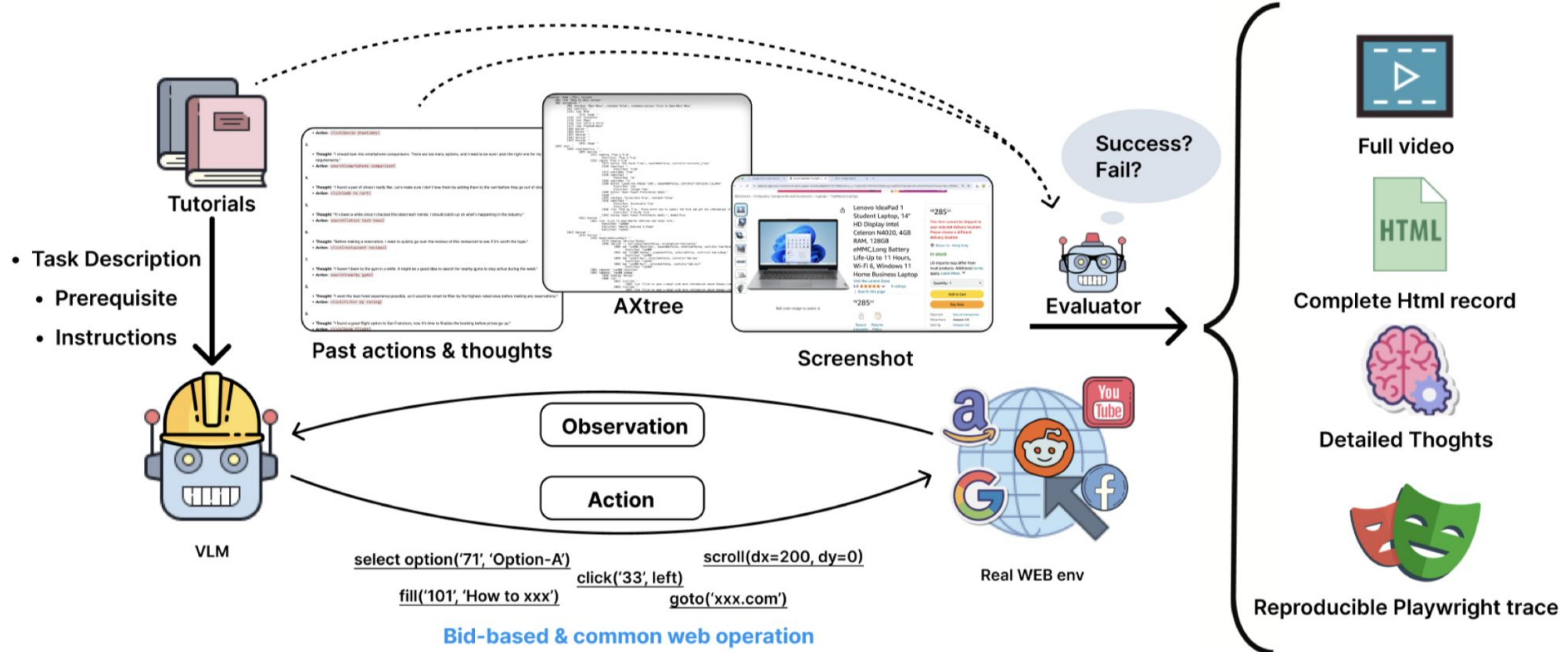
```
The following is a tutorial from the website.  It may contain several
tutorials.  Please extract the first tutorial only and format the
first tutorial according to the specified schema:

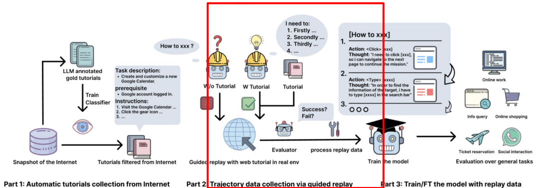
Text:  {context}

Schema:
{
    "platform":
    "Platform category (choose from:  macOS, Windows (Default if not
specified in the tutorial), Linux, Android, iOS)",
    "target type":
    "Type of platform (choose from:  Web browser, PC app, Mobile app,
PC operating system, Mobile operating system, where the tutorial's
steps are performed).  Tutorials that involve interacting with the
browser software itself, such as 'opening Chrome settings,' should be
classified as a PC app type.",
    "target object":
    "Specific name of the web browser or (non web browser)
applications or operating system where the tutorial's steps are
performed (e.g., Chrome browser (Default for browser and web
tutorial), Microsoft Excel (app name), Windows system settings)",
    "target web URL":
    "The exact URL of the web page where the tutorial's actions take
place, applicable only if the target object is a web browser (e.g.,
None, https://mail.google.com, https://www.amazon.com,
https://github.com).  Be careful, the URL provided at the beginning
is always not the URL where the tutorial's actions are about.  For
example, a tutorial from
https://abidakon.com/how-to-make-google-slide-vertical/ about
changing Google Slides, its target web URL should be
https://docs.google.com/presentation.",
    "task description":
    "Task description text (Provide a concise summary in one
sentence, including essential details)",
    "prerequisites":
     "Prerequisite text describing necessary conditions before
starting the task",
    "instructions":
    [
     "Step_1:  Instruction text describing the action to be taken",
     // Following instructions
    ]
    "instructions steps":
    "Total number of instructions steps",
    "expected_result":
    "Text describing the expected result after following the
instructions"
}
```

- **Platform and Target Environment**: Specifies the operating system, software version, and relevant dependencies.
- **Task Description**: Provides a concise problem statement that defines the objective of the task.
- **Prerequisites**: Lists necessary dependencies, tools, and background knowledge required to complete the task.
- **Step-by-Step Instructions**: Offers procedural guidance, including command syntax and sequential actions.
- **Expected Outcome:** Defines the anticipated results or outputs upon successful task completion.

Trajectory Synthesis via Guided Replay

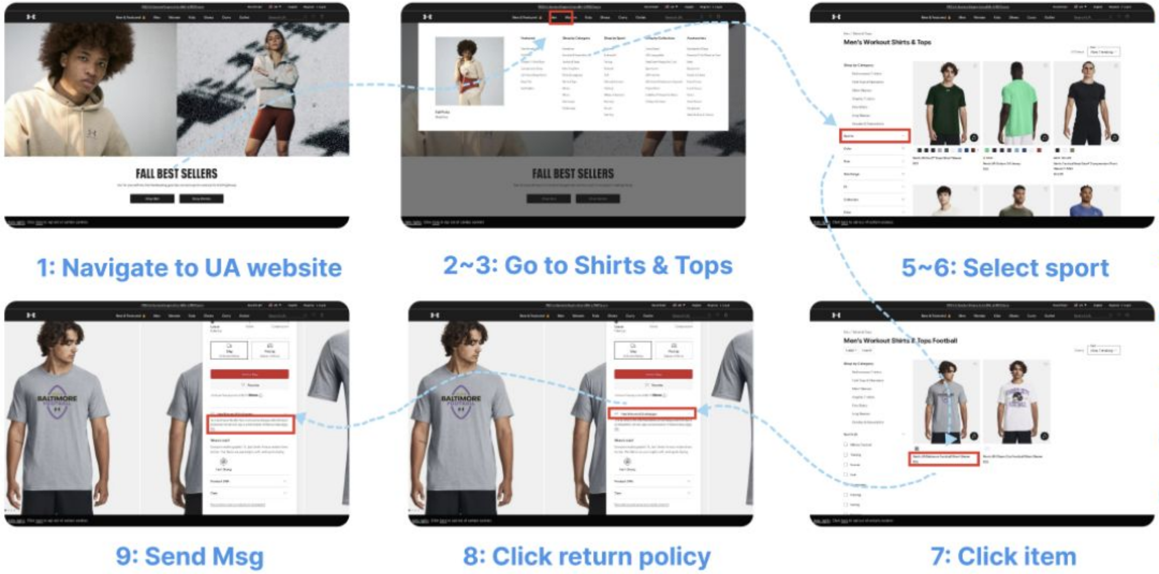




Example

salesforce

Task: Find the return policy for any men's football apparel on Under Armour's website.



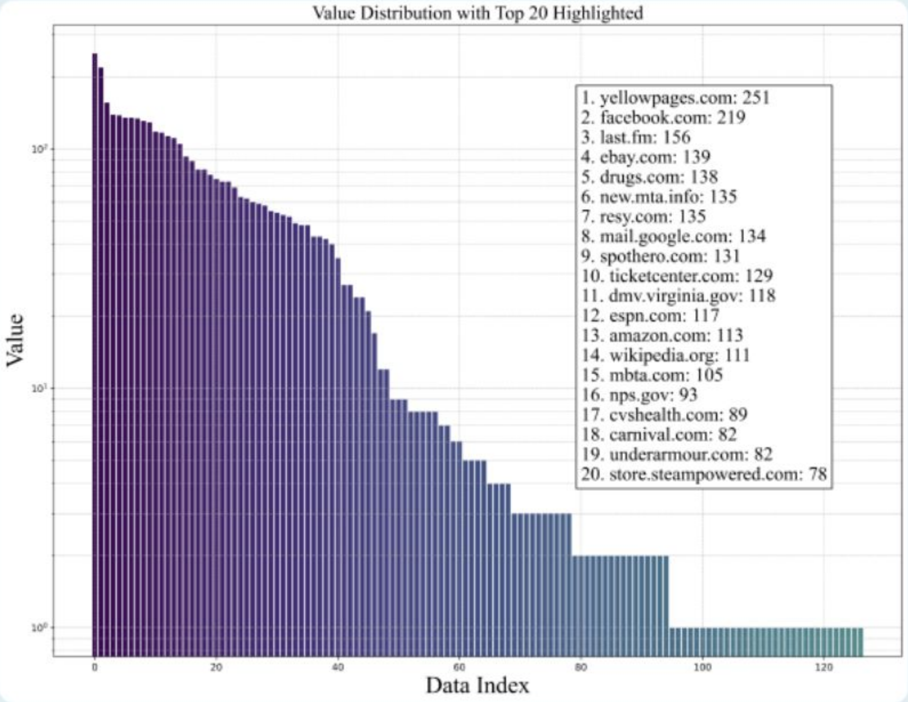
Thoughts
Actions

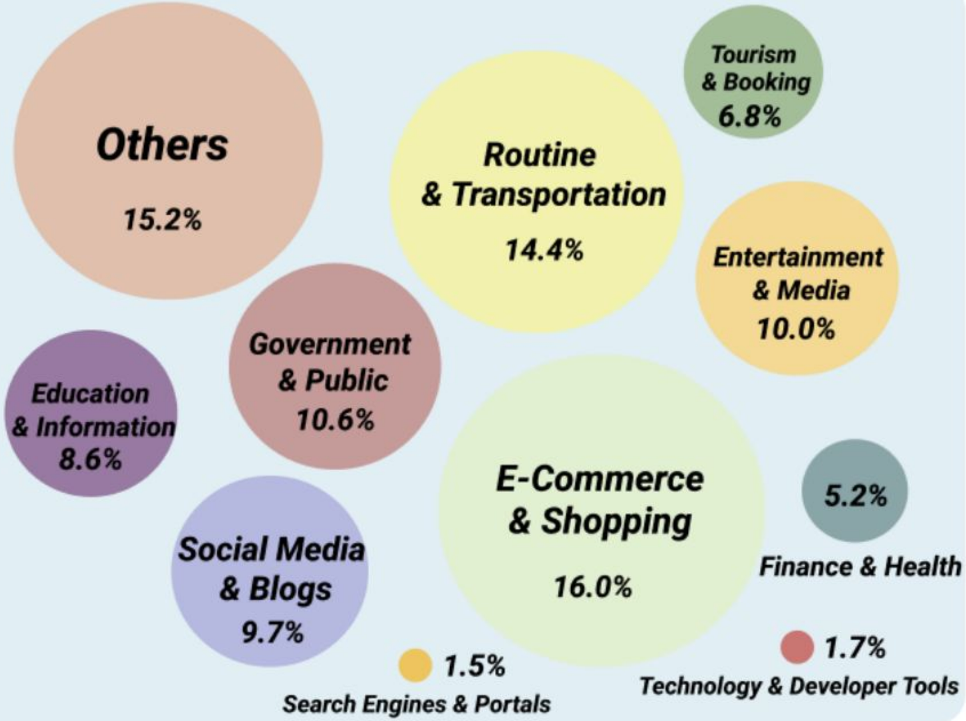
1. 'I start by navigating to the Under Armour website.'
   goto('https://www.underarmour.com')
2. 'Next, I hover over the "Men" menu to bring up the dropdown.'
   hover('250')
3. 'I proceed by clicking on "Shirts & Tops" from the dropdown.'
   click('295')
4. 'To continue, I close the dialog that appears.'
   click('122')
5. 'I then locate and click on the "Sports" section.'
   click('2632')
6. 'After that, I click on the "Football" link to move forward.'
   click('2662')
7. 'Pick a product to check out its details and find the return policy.'
   click('4969')
8. 'I click the "Free Returns & Exchanges" button to view the policy.'
   click('5629')
9. 'Now, I can see the return policy information on the page.'
   send_msg_to_user("Under Armour offers free returns and exchanges within 60 days...")

Benchmark Comparison

Datasets	Size	Average Steps	HTML	AxTree	Intermediate Reasoning	Video	Matching Screenshot	Website	Task Inst. Level
RUSS	80	5.4	Yes	No	No	No	No	22	Low
ScreenAgent	203	4.3	No	No	Yes	No	Yes	-	High & Low
WebLINX	969	18.8	Yes	No	No	No	Yes	155	High & Low
MM-Mind2Web	1009	7.3	Yes	No	No	No	No	137	High
GUIAct	2482	6.7	No	No	No	No	Yes	121	High
AgentTrek (Ours)	**10398**	**12.1**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	127	**High & Low**

Distribution of websites and domains





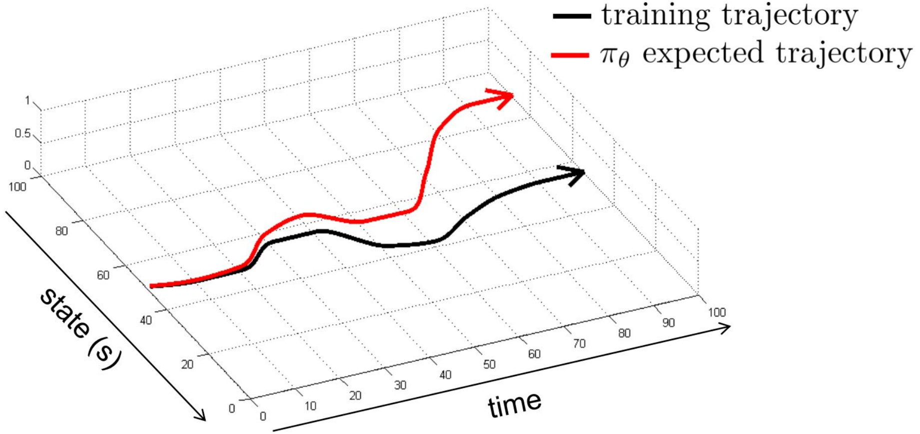
Comparison on WebArena

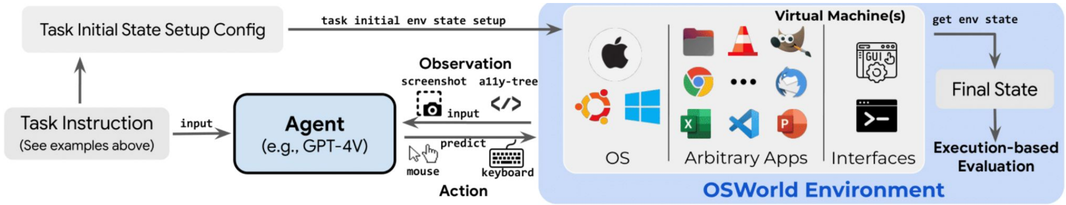
Model	WebArena
LLaMa3-chat-8B (Ou et al., 2024)	3.32
Qwen2.5-7B-Instruct	3.80
LLama3-chat-70B (Ou et al., 2024)	7.02
GPT-4o (Zhou et al., 2023)	13.10
GPT-4 (Ou et al., 2024)	14.41
Synatra-CodeLlama-7B (Ou et al., 2024)	6.28
AutoWebGLM (OOD SFT) (Lai et al., 2024)	8.50
Qwen2.5-7B-Instruct w/ **AgentTrek**	10.46
Qwen2.5-32B-Instruct w/ **AgentTrek**	**22.40**

Takeaways

- Diverse task source with knowledge.
- Realistic trajectories with reasoning and reflection.
- Scalable data synthesis to reuse web corpus.

- However, from imitation learning to reinforcement learning in environment.
- Combine with **OSWorld** (SFT→RL)





Does Multimodal LLM itself need action call capability?

If does, can we crawl from web data?

Open-source MLLMs still fail at difficult/complex questions

Q: What is the price for tomatoes?
A: 8.0

Mantis-LLaVA: 1.5
LLaVA-OV: 7.00

Finegrained OCR

Q: How many kids are in front of the yellow schoolbus?
A. 5; B. 4; C. 3; D. 6
A: B

Mantis-LLaVA: C
LLaVA-OV: A. There are 5 kids in front of the yellow schoolbus.

Visual grounding & counting

Q: How many gallons of supreme gasoline can I get with $50?
A: 13.7

Mantis-LLaVA: 3.6
LLaVA-OV: 5.2

Multi-step recog. & reasoning

Q: Can you give a short introduction to this painting?
A: The Starry Night is an oil-on-canvas painting by Vincent van Gogh that depicts the view from the east-facing window of his asylum room at Saint-Rémy-de-Provence.

Mantis-LLaVA: The painting is a depiction of a starry night sky with a large starry sky in the background.
LLaVA-OV: Starry Night by Vincent van Gogh.

External knowledge

We introduce 🌮TACO: Multi-modal Action Models with Synthetic Chains-of-Thought-and-Action (CoTA)

Q: What is the price for tomatoes?
A: 8.0

Mantis-LLaVA: 1.5
LLaVA-OV: 7.00

🌮 TACO

t_0: To determine the price of tomatoes, I need to extract the text...
a_0: **OCR**(image-0)
o_0: “1. DOMATES TINAZTEPE - 8,00...11. ANTEP FISTIKI INCIK - 15,00 These appear to be labels for various fruits and vegetables with their prices.”

t_1: The price for tomatoes is visible in the extracted text as “DOMATES TINAZTEPE” with a price of “8,00”.
a_1: **Terminate**(“8.00”)

Q: How many kids are in front of the yellow schoolbus?
A. 5; B. 4; C. 3; D. 6
A: B

Mantis-LLaVA: C

LLaVA-OV: A. There are 5 kids in front of the yellow schoolbus.

salesforce

🌮 **TACO**

t_0: To determine the number of kids in front of the yellow school bus, I need to analyze the image for people in that area.
a_0: **LocalizeObjects**(image-0, “kid in front of the bus”)
o_0: [{'label': 'person', 'bbox': [0.79, 0.59, 0.85, 0.81], 'score': 0.62},
{'label': 'person-2', 'bbox': [0.85, 0.58, 0.92, 0.83], 'score': 0.55},
{'label': 'person-3', 'bbox': [0.92, 0.6, 0.98, 0.81], 'score': 0.51},
{'label': 'person-4', 'bbox': [0.91, 0.6, 0.94, 0.8], 'score': 0.43}]

t_1: The image shows 4 people in front of the school bus. This matches option B.
a_1: **Terminate**(“B”)

Q: Can you give a short introduction to this painting?
A: The Starry Night is an oil-on-canvas painting by Vincent van Gogh that depicts the view from the east-facing window of his asylum room at Saint-Rémy-de-Provence.

Mantis-LLaVA: The painting is a depiction of a starry night sky with a large starry sky in the background.

LLaVA-OV: Starry Night by Vincent van Gogh.

TACO

t_0: I can provide information about the painting based on general knowledge.
a_0: **QueryKnowledgeBase**(“Starry Night painting by Vincent van Gogh”)
o_0: {“title”: “The Starry Night”, “url”: “https://en.wikipedia.org/wiki/The_Starry_Night”, “summary”: The Starry Night is an oil-on-canvas painting by the Dutch Post-Impressionist painter Vincent van Gogh, painted in June 1889. It depicts the view from the east-facing window of his asylum room at Saint-Rémy-de-Provence, just before sunrise, with the addition of an imaginary village.”}

t_1: I have enough information to provide a short introduction to the painting.
a_1: **Terminate**(“The Starry Night is an oil-on-canvas painting by Vincent van Gogh, depicting a view from his asylum room at Saint-Rémy-de-Provence, with an imaginary village and a starry sky.”)

Q: How many gallons of supreme gasoline can I get with $50?
A: 13.7

Mantis-LLaVA: 3.6
LLaVA-OV: 5.2

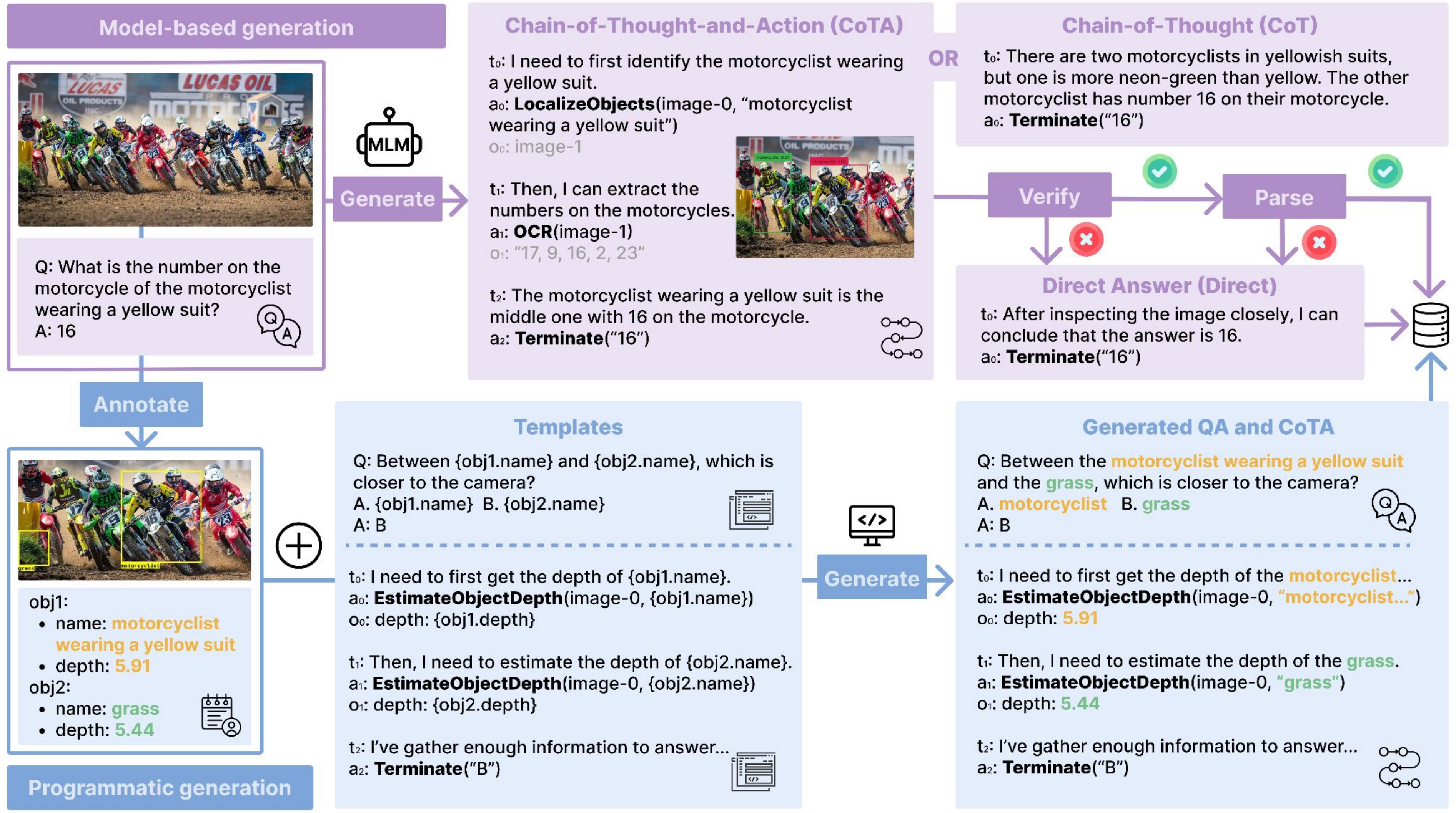
TACO

t_0: I need to extract the price of supreme gasoline from the image.
a_0: **OCR**(image-0)
o_0: “Gasoline - Self Serve - Regular - 3.49 9/10 - Plus - 3.55 9/10 - Supreme - 3.65 9/10 - Diesel No. 2 - 3.60 9/10”

t_1: The price of supreme gasoline is $3.65 per gallon...I need to divide $50 by $3.65.
a_1: **Calculate**(“50 / 3.65”)
o_1: 13.6986

t_2: The calculation shows that you can get approximately 13.7 gallons of supreme gasoline with $50.
a_2: **Terminate**(“13.7”)

Synthetic CoTA Generation Pipeline



Model-based generation
Q: What is the number on the motorcycle of the motorcyclist wearing a yellow suit?
A: 16
MLM
Generate
Chain-of-Thought-and-Action (CoTA)
t_0: I need to first identify the motorcyclist wearing a yellow suit.
a_0: LocalizeObjects(image-0, "motorcyclist wearing a yellow suit")
o_0: image-1
t_1: Then, I can extract the numbers on the motorcycles.
a_1: OCR(image-1)
o_1: "17, 9, 16, 2, 23"
t_2: The motorcyclist wearing a yellow suit is the middle one with 16 on the motorcycle.
a_2: Terminate("16")
OR
Chain-of-Thought (CoT)
t_0: There are two motorcyclists in yellowish suits, but one is more neon-green than yellow. The other motorcyclist has number 16 on their motorcycle.
a_0: Terminate("16")
Verify
Parse
Direct Answer (Direct)
t_0: After inspecting the image closely, I can conclude that the answer is 16.
a_0: Terminate("16")
Annotate
obj1:
• name: motorcyclist wearing a yellow suit
• depth: 5.91
obj2:
• name: grass
• depth: 5.44
Programmatic generation
Templates
Q: Between {obj1.name} and {obj2.name}, which is closer to the camera?
A. {obj1.name} B. {obj2.name}
A: B
t_0: I need to first get the depth of {obj1.name}.
a_0: EstimateObjectDepth(image-0, {obj1.name})
o_0: depth: {obj1.depth}
t_1: Then, I need to estimate the depth of {obj2.name}.
a_1: EstimateObjectDepth(image-0, {obj2.name})
o_1: depth: {obj2.depth}
t_2: I've gather enough information to answer...
a_2: Terminate("B")
Generate
Generated QA and CoTA
Q: Between the motorcyclist wearing a yellow suit and the grass, which is closer to the camera?
A. motorcyclist B. grass
A: B
t_0: I need to first get the depth of the motorcyclist...
a_0: EstimateObjectDepth(image-0, "motorcyclist...")
o_0: depth: 5.91
t_1: Then, I need to estimate the depth of the grass.
a_1: EstimateObjectDepth(image-0, "grass")
o_1: depth: 5.44
t_2: I've gather enough information to answer...
a_2: Terminate("B")

Templates for programmatic data generation

# of input images	Capabilities	Question Template	Action Template
1	Counting	How many {object} are there? Among {objects}, which is the most frequent object? Among {objects}, which object appears the least?	LocalizeObjects
	Counting, Attribute recognition	How many {attribute} {object} are there?	
	2D spatial reasoning	Among {objects}, which is on the most left side? Among {objects}, which is on the most right side? Among {objects}, which is on the most top side? Among {objects}, which is on the most bottom side?	
	3D spatial reasoning	Which of {objects} is closer? Which of {objects} is farther?	LocalizeObjects, EstimateRegionDepth x2 OR, EstimateObjectDepth x2
2-3	Multi-image understanding Multi-image understanding, Counting Multi-image understanding, Counting Multi-image understanding, Counting Multi-image understanding, Attribute recognition Multi-image understanding, Attribute recognition, Counting	Which image has {object}? How many {object} are in in these images? Which image has most {object}? Which image has least {object}? Which image has {attribute} {object}? How many {attribute} {object} in these images?	LocalizeObjects x N

Action Set: OCR, GETOBJECTS, LOCALIZEOBJECTS, ESTIMATEOBJECTDEPTH, ESTIMATEREGIONDEPTH, GETIMAGETOTEXTSSIMILARITY, GETIMAGETOIMAGESSIMILARITY, GETTEXTTOIMAGESSIMILARITY, DETECTFACES, CROP, ZOOMIN, QUERYLANGUAGEMODEL, QUERYKNOWLEDGEBASE, CALCUATE, and SOLVEMATHEQUATION.

1. CoTA finetuning elicits multi-modal models' reasoning and action calling abilities and significantly boosts their performance, which few-shot prompting fails to achieve.

Table 1. **CoTA inference before vs. after fine-tuning.** While GPT-4o performs well with either a direct answer (Direct) or chain-of-thought-and-action (CoTA) prompt, open-source multi-modal models lag behind and fail to generate CoTA with few-shot prompting. We show that fine-tuning with CoTA data elicits their reasoning and action calling abilities and significantly boosts their performance.

Model	Language / Vision backbone	Train data / Inference format	A-OKVQA	BLINK	MathVista	MMMU	MMStar	MMVet	MMVP	RealWorldQA	Avg
GPT-4o (2024-08-06)	—	— / Direct	88.4	64.7	60.5	67.6	64.5	70.0	84.7	72.0	71.5
		— / CoTA	89.9	63.2	59.0	64.6	64.3	67.2	83.0	69.9	70.1
GPT-4o (language-only)		— / CoTA	74.8	45.6	44.5	54.1	55.3	45.2	58.0	50.2	53.5
Mantis-instruction-tuned	LLaMA3-8B / SigLIP	— / Direct	81.2	46.4	34.4	40.1	40.1	36.9	69.0	51.0	49.9
		— / CoTA	0.5	0.0	20.0	1.5	1.7	0.0	0.0	0.0	3.0
TACO		CoTA 293K / CoTA	81.8	47.6	36.3	40.9	42.5	45.7	65.3	56.5	52.1
LLaVA-OV-Stage1.5	Qwen2-7B / SigLIP	— / Direct	76.1	34.8	35.9	36.1	39.1	32.3	63.7	54.1	46.5
		— / CoTA	25.7	8.8	21.5	21.2	26.7	7.2	40.5	37.5	23.6
TACO		CoTA 293K / CoTA	**85.9**	**49.9**	**41.9**	**44.0**	**51.0**	**50.9**	**72.3**	**58.8**	**56.8**

2. Our best CoTA data recipe enables TACO to consistently beat instruction-tuned baselines by 1-4% on average across 8 benchmarks, with significant gains of up to 15% on MMVet.

Table 2. **Best CoTA data recipe.** Chain-of-Thought-and-Action (CoTA) data improves models' average performance across 8 multi-modal benchmarks by 1-4% compared to instruction tuning data of the same examples with only direct answers (Direct). We use colors to highlight whether CoTA data increases or decreases performance on a particular benchmark compared to the instruction-tuned baseline.

Model	Language / Vision backbone	Start checkpoint / Seen data	Train data / Inference format	A-OKVQA	BLINK	MathVista	MMMU	MMStar	MMVet	MMVP	RealWorldQA	Avg	Delta
Mantis	LLaMA3-8B / CLIP	Pretrained / 558K	Direct 293K / Direct	80.7	45.8	33.1	42.2	36.7	28.9	62.7	52.3	47.8	
TACO			CoTA 293K / CoTA	81.1	49.6	36.6	42.8	40.8	45.2	63.3	51.1	51.3	+3.5
Mantis	LLaMA3-8B / SigLIP	Pretrained / 558K	Direct 293K / Direct	80.3	43.7	31.1	40.4	40.5	33.0	63.3	51.8	48.0	
TACO			CoTA 293K / CoTA	82.4	47.8	34.9	40.3	44.6	45.2	64.0	53.7	51.6	**+3.6**
Mantis		Instruction tuned / 1.2M	Direct 293K / Direct	81.1	46.7	36.2	40.7	40.7	29.7	68.3	54.8	49.8	
TACO			CoTA 293K / CoTA	81.8	47.6	36.3	40.9	42.5	45.7	65.3	56.5	52.1	+2.3
LLaVA-OV	Qwen2-7B / SigLIP	Stage 1 / 558K	Direct 293K / Direct	83.1	49.5	38.4	45.6	42.3	33.0	69.7	55.3	52.1	
TACO			CoTA 293K / CoTA	84.5	49.6	41.8	45.3	44.5	48.9	66.7	53.6	54.4	+2.3
LLaVA-OV		Stage 1.5 / 4.5M	Direct 293K / Direct	85.5	**50.3**	**42.4**	**46.1**	50.1	39.3	**73.6**	57.8	55.6	
TACO			CoTA 293K / CoTA	**85.9**	49.9	41.9	44.0	**51.0**	**50.9**	72.3	**58.8**	**56.8**	+1.2

3. Quality >> quantity: a) the smallest CoTA dataset results in better average performance and higher gains compared to larger datasets with a mix of CoTA, CoT and/or Direct examples.

Table 3. **Model-generated data ablations.** Data quality matters more than quantity. We find that (1) the smallest dataset with only CoTA examples results in better average performance and higher gains compared to other larger datasets with a mix of CoTA and Direct examples; (2) filtering out Action-useless datasets leads to performance gains.

Data source	Final data format	Size	Model	A-OKVQA	BLINK	MathVista	MMMU	MMStar	MMVet	MMVP	RealWorldQA	Avg	Delta
All datasets	Direct	293K	Mantis-SigLIP	80.3	43.7	31.1	40.4	40.5	33.0	63.3	51.8	48.0	
	CoTA		TACO	82.4	**47.8**	34.9	40.3	**44.6**	**45.2**	64.0	53.7	**51.6**	**+3.6**
	Direct	580K	Mantis-SigLIP	82.3	45.2	34.2	**42.6**	39.5	31.9	**67.7**	52.6	49.5	
	CoTA+ CoT		TACO	**84.0**	46.4	**36.3**	40.3	40.6	43.7	66.7	51.6	51.2	+1.7
	Direct	528K	Mantis-SigLIP	81.7	47.1	35.0	39.7	40.5	27.1	65.3	52.3	48.6	
	CoTA+ Direct		TACO	80.5	43.3	35.7	37.2	40.9	40.2	50.0	50.9	47.3	-1.3
	Direct	815K	Mantis-SigLIP	82.5	46.1	34.4	40.5	40.2	29.9	65.7	**55.0**	49.3	
	CoTA+ CoT+ Direct		TACO	81.6	44.9	34.1	40.5	39.5	30.8	62.0	48.5	47.7	-1.6
Action-useful datasets	Direct	566K	Mantis-SigLIP	81.0	41.2	32.7	41.9	40.3	26.2	66.0	49.5	47.4	
	CoTA+ CoT+ Direct		TACO	82.5	42.2	32.4	42.5	40.7	34.3	64.7	47.7	48.4	+1.0

3. Quality >> quantity: b) filtering out Action-useless datasets also leads to performance gains.

Table 3. **Model-generated data ablations.** Data quality matters more than quantity. We find that (1) the smallest dataset with only CoTA examples results in better average performance and higher gains compared to other larger datasets with a mix of CoTA and Direct examples; (2) filtering out Action-useless datasets leads to performance gains.

Data source	Final data format	Size	Model	A-OKVQA	BLINK	MathVista	MMMU	MMStar	MMVet	MMVP	RealWorldQA	Avg	Delta
	Direct	293K	Mantis-SigLIP	80.3	43.7	31.1	40.4	40.5	33.0	63.3	51.8	48.0	
	CoTA		TACO	82.4	**47.8**	34.9	40.3	**44.6**	**45.2**	64.0	53.7	**51.6**	**+3.6**
All datasets	Direct	580K	Mantis-SigLIP	82.3	45.2	34.2	**42.6**	39.5	31.9	**67.7**	52.6	49.5	
	CoTA+ CoT		TACO	**84.0**	46.4	**36.3**	40.3	40.6	43.7	66.7	51.6	51.2	+1.7
	Direct	528K	Mantis-SigLIP	81.7	47.1	35.0	39.7	40.5	27.1	65.3	52.3	48.6	
	CoTA+ Direct		TACO	80.5	43.3	35.7	37.2	40.9	40.2	50.0	50.9	47.3	-1.3
	Direct	815K	Mantis-SigLIP	82.5	46.1	34.4	40.5	40.2	29.9	65.7	**55.0**	49.3	
	CoTA+ CoT+ Direct		TACO	81.6	44.9	34.1	40.5	39.5	30.8	62.0	48.5	47.7	-1.6
Action-useful datasets	Direct	566K	Mantis-SigLIP	81.0	41.2	32.7	41.9	40.3	26.2	66.0	49.5	47.4	
	CoTA+ CoT+ Direct		TACO	82.5	42.2	32.4	42.5	40.7	34.3	64.7	47.7	48.4	+1.0

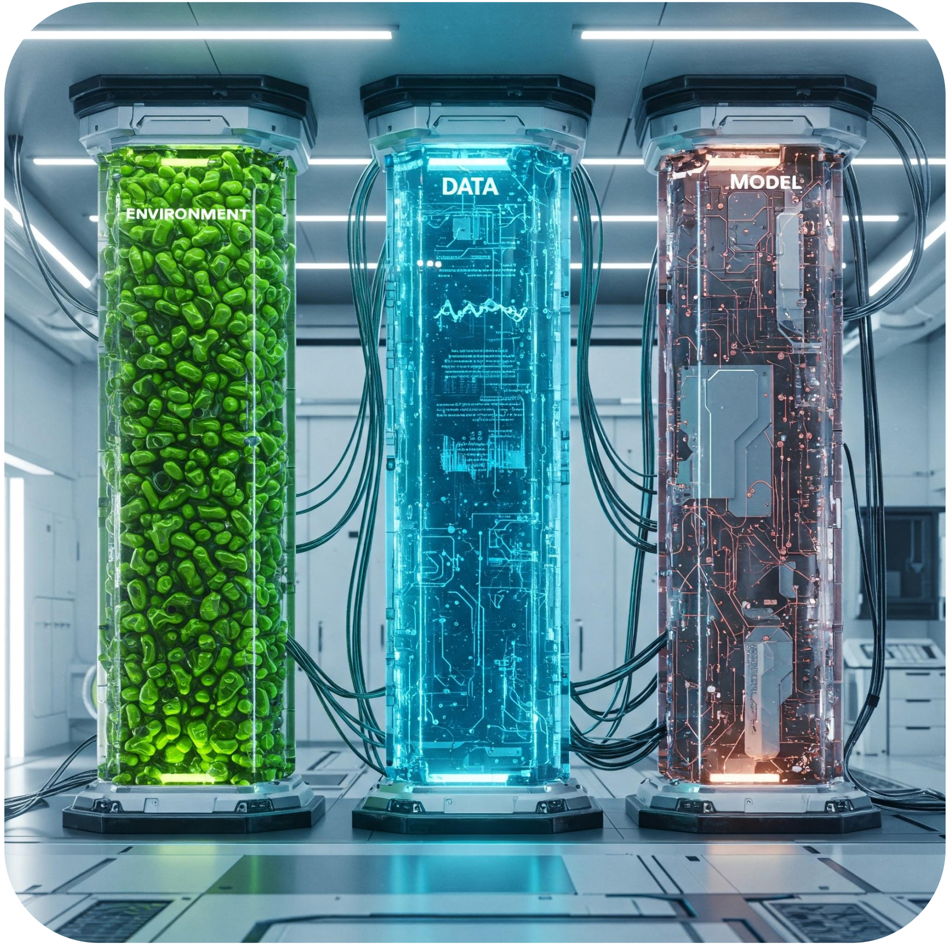
4. Adding programmatic data can bring gains on some benchmarks but not on the average performance.

Table 4. **Model-generated and program-generated data mixtures.** Adding programmatically generated CoTA data can increase the model's performance on some benchmarks such as A-OKVQA, MathVista, MMMU, and MMVP. However, it doesn't further improve model's average performance across all benchmarks. Additionally, more programmatic CoTA can even hurt the model's performance. M:P = Model-generated CoTA (M-CoTA): Program-generated CoTA (P-CoTA).

Model	M:P	Train data	Total size	A-OKVQA	BLINK	MathVista	MMMU	MMStar	MMVet	MMVP	RealWorldQA	Avg	Delta
Mantis-SigLIP		Direct 293K	293K	80.3	43.7	31.1	40.4	40.5	33.0	63.3	51.8	48.0	
TACO	M:P	M-CoTA / P-CoTA											
	0:1	0 / 293K	293K	34.3	37.4	17.3	31.9	30.4	0.0	48.3	40.7	30.0	—
	1:0	293 / 0K	293K	82.4	**47.8**	34.9	40.3	**44.6**	**45.2**	64.0	**53.7**	**51.6**	**+3.6**
	1:0.1	293 / 29K	322K	**82.6**	47.5	33.9	40.3	44.2	42.3	64.3	49.8	50.6	+2.6
	1:0.25	293 / 73K	366K	82.1	44.2	**38.3**	40.2	42.9	45.1	**64.7**	51.2	51.1	+3.1
	1:0.5	293 / 147K	440K	81.9	46.0	36.7	**41.4**	41.4	40.9	62.3	50.3	50.1	+2.1
	1:1	293 / 293K	586K	81.1	47.7	31.0	39.3	41.4	36.2	63.0	50.7	48.8	+0.8

Takeaways

- Action call capability should be the default ability in MLLMs.
- CoTA finetuning >> few-shot CoTA.
- CoTA data consistently improves baselines trained on instruction-tuning data with only direct answers.
- CoTA quality >> quantity.



Generated from Gemini

Agenda

01 — Environment/Benchmark: Should be reconfigurable and expandable

02 — Data: Diverse modalities, large-scale trajectory data, covering a wide range of tasks

03 — **Model/System: Unified vision-language-reasoning-acti on model, and long-context inference.**

Aguvis: Unified Pure Vision Agents for Autonomous GUI Interaction

Background

- Heterogeneous textual GUI interface representation
- Limited visual grounding capability
- Perform “reactive” low-level actions directly without reasoning

Observation: Complex and different textual representation for GUI interface.

```html
<html>
<head>
    <title>
        Galaxy S20
    </title>
</head>
<body>
<div>
    <li>
        <div>
            <span> Display </span>
            <span> 6.5 inch </span>
        </div>
    </li>
    <li>
        <div>
            <span> Processor </span>
            <span> Qualcomm Snapdragon </span>
        </div>
    </li>
    <ul>
        <li>
            <span> Release Date </span>
            2020
        </li>
    </ul>
</div>
</body>
</html>
```

Web (HTML)

OS (AXTree)


```json
{
  "activity_name": "com.funforfones.android.chicagocta/com.funforfones.android.chicagocta.Ma
  "activity": {
    "root": {
      "scrollable-horizontal": false,
      "draw": true,
      "ancestors": [
        "android.widget.FrameLayout",
        "android.view.ViewGroup",
        "android.view.View",
        "java.lang.Object"
      ],
      "clickable": false,
      "pressed": "not_pressed",
      "focusable": false,
      "long-clickable": false,
      "enabled": true,
      "bounds": [
        36,
        84,
        1404,
        2392
      ],
      "visibility": "visible",
      "content-desc": [
        null
      ],
      "rel-bounds": [
        0,
        0,
        1368,
        2308
      ],
      "focused": false,
      "selected": false,
      "scrollable-vertical": false,
      "children": [
          ....
      ],
      "adapter-view": false,
      "abs-pos": true,
      "pointer": "2e18ce7",
      "class": "com.android.internal.policy.PhoneWindow$DecorView",
      "visible-to-user": true
    },
    "added_fragments": [],
    "active_fragments": []
  },
  "is_keyboard_deployed": true,
  "request_id": "1350"
}
```


Mobile (XML)

Different observation representations result in different action grounding spaces, even on the same platform.

Action Type
click
hover
type
select

Simplified Browser API (Mind2Web)

Action Type a
click [elem]
hover [elem]
type [elem] [text]
press [key_comb]
new_tab
tab_focus [index]
tab_close
goto [url]
go_back
go_forward
scroll [up|down]
stop [answer]

Enhanced Browser API (VisualWebArena)

Category	Primitive
bid	fill(bid, text) click(bid, button) dblclick(bid, button) hover(bid) press(bid, key_comb) focus(bid) clear(bid) select_option(bid, options) drag_and_drop(from_bid, to_bid)
coord	mouse_move(x, y) mouse_down(x, y, button) mouse_up(x, y, button) mouse_click(x, y, button) mouse_dblclick(x, y, button) mouse_drag_and_drop(from_x, from_y, to_x, to_y) keyboard_down(key) keyboard_up(key) keyboard_press(key_comb) keyboard_type(text) keyboard_insert_text(text)
tab	new_tab() tab_close() tab_focus(index)
nav	go_back() go_forward() goto(url)
misc	scroll(dx, dy) send_msg_to_user(text) noop()
python	Any python code (UNSAFE!)

Playwright Browser HTML-based API

Function
moveTo(x, y)
click(x, y)
write('text')
press('enter')
hotkey('ctrl', 'c')
scroll(200)
scroll(-200)
dragTo(x, y)
keyDown('shift')
keyUp('shift')
WAIT
FAIL
DONE

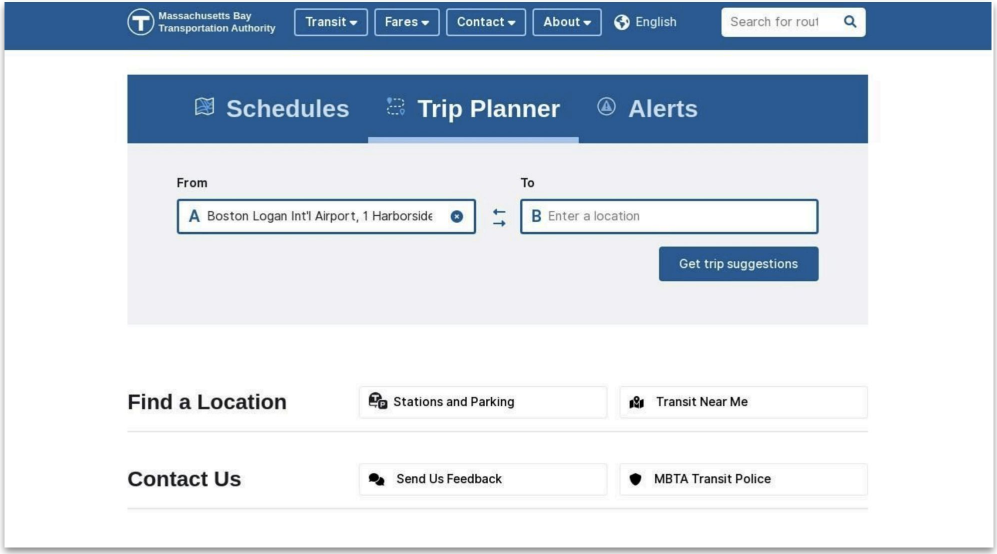
PyAutoGUI OS Vision-based API

Limited visual grounding capability

Grounder	Mobile		Desktop		Web		Avg
	Text	Icon/Widget	Text	Icon/Widget	Text	Icon/Widget	
GPT-4	22.6	24.5	20.2	11.8	9.2	8.8	16.2
GPT-4o	20.2	24.9	21.1	23.6	12.2	7.8	18.3
CogAgent	67.0	24.0	74.2	20.0	70.4	28.6	47.4
SeeClick	78.0	52.0	72.2	30.0	55.7	32.5	53.4
Qwen2-VL	75.5	60.7	76.3	54.3	35.2	25.7	55.3

Perform "reactive" low-level actions directly without reasoning

Image Input



Prompt

Please generate the next move according to the UI screenshot, instruction and previous actions.

Instruction: Plan a trip from Boston Logan Airport to North Station.

Previous actions:

Step 1: pyautogui.click(x=0.4754, y=0.2062)

Step 2: pyautogui.click(x=0.3295, y=0.4)

pyautogui.write(text='Boston Logan Airport')

Step 3: pyautogui.click(x=0.3262, y=0.4764)

Generation

Action:

pyautogui.click(x=0.6756, y=0.4)

pyautogui.write(text='North Station')

Aguvis: Unified Pure Vision Agents for Autonomous GUI Interaction

- Heterogeneous textual GUI interface representation

  → **Unified Vision-based perception and action space for GUI Interaction**

- Limited visual grounding capability

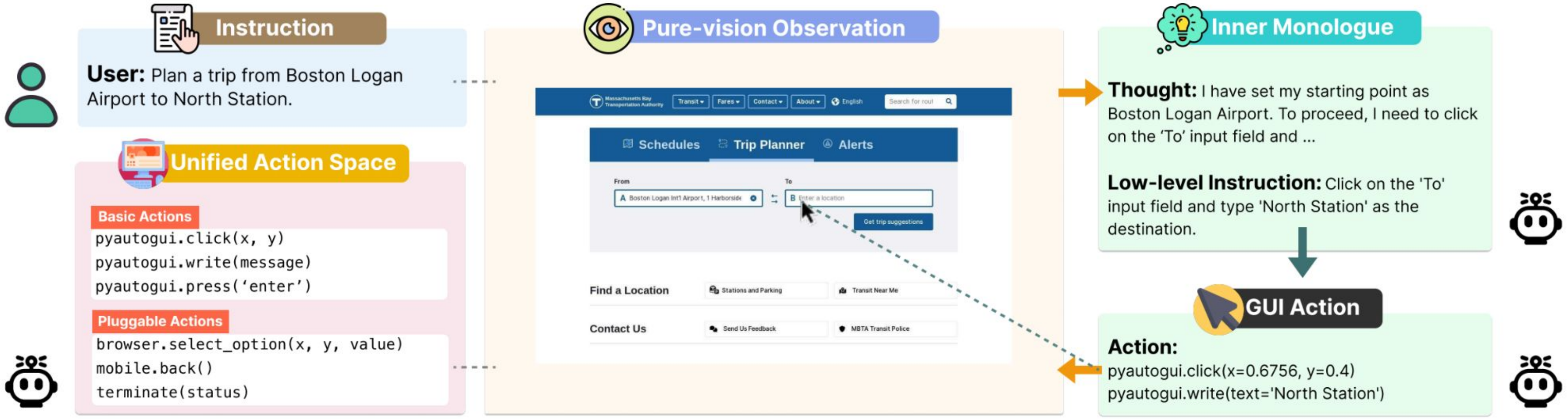
  → **Improving visual action grounding capability through training**

- Perform “reactive” low-level actions directly without reasoning

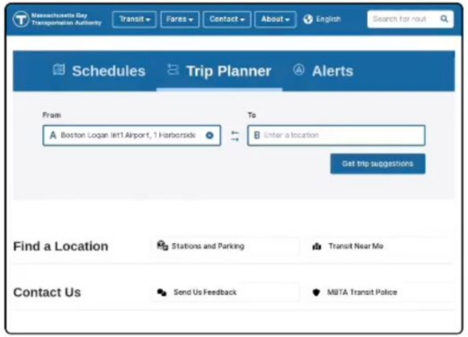
  → **Explicit reasoning process / inner monologue**

Aguvis: Unified Pure Vision Agents for Autonomous GUI Interaction (Xu et. al, 2024)

Aguvis: Unified Pure Vision Agents for Autonomous GUI Interaction



Observation



Instruction

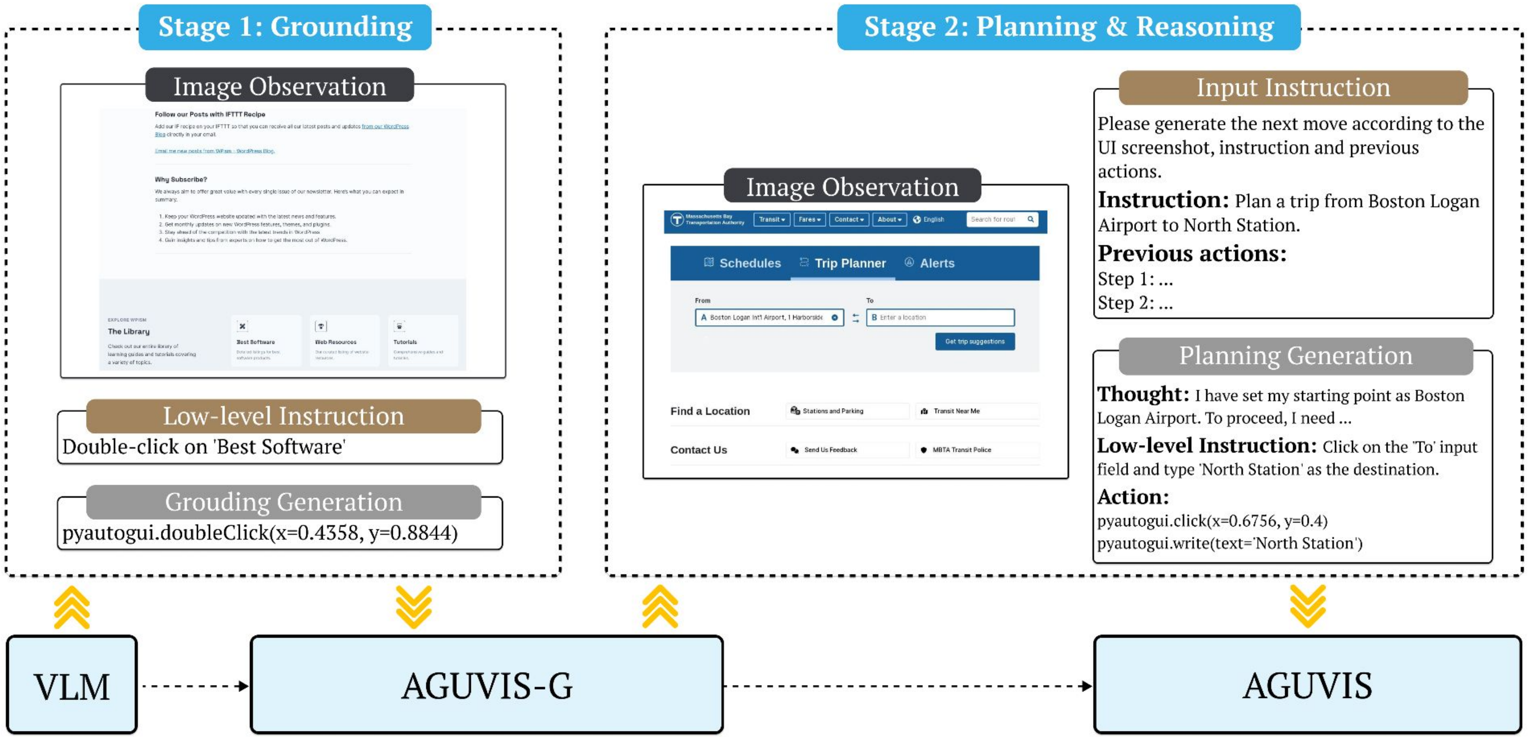
Please generate the next move according to the UI screenshot, instruction and previous actions.
Instruction: Plan a trip from Boston Logan Airport to North Station.
Previous actions:
Step 1: ...
Step 2: ...

→

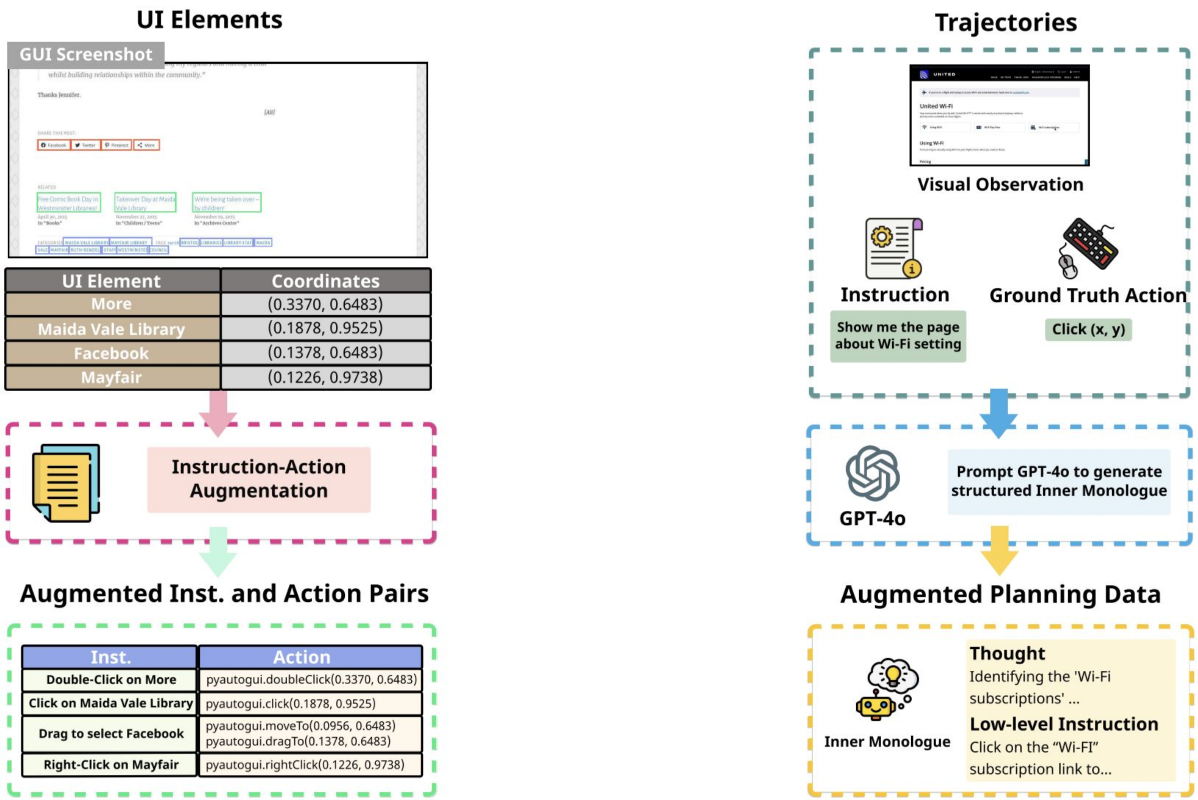
Inner Monologue and Action

Thought: I have set my starting point as Boston Logan Airport. To proceed, I need ...
Low-level Instruction: Click on the 'To' input field and type 'North Station' as the destination.
Action:
pyautogui.click(x=0.6756, y=0.4)
pyautogui.write(text='North Station')

Two-Stage Training



Data Collection Pipeline



Example of Inner monologue Augmentation

salesforce

Prompt

Please generate the next move according to the UI screenshot, instruction and previous actions.

Instruction: Plan a trip from Boston Logan Airport to North Station.

Previous actions:
Step 1: pyautogui.click(x=0.4754, y=0.2062)
Step 2: pyautogui.click(x=0.3295, y=0.4)
pyautogui.write(text='Boston Logan Airport')
Step 3: pyautogui.click(x=0.3262, y=0.4764)

Generation

Action:
pyautogui.click(x=0.6756, y=0.4)
pyautogui.write(text='North Station')

Prompt

Please generate the next move according to the UI screenshot, instruction and previous actions.

Instruction: Plan a trip from Boston Logan Airport to North Station.

Previous actions:
Step 1: Click on the 'Trip Planner' tab to begin planning the trip.
Step 2: Click on the 'From' input field and type 'Boston Logan Airport'.
Step 3: Click on 'Boston Logan Int'l Airport, 1 Harborside Dr, East Boston, MA 02128, United States' to set it as my starting location.

Generation

Thought: I have set my starting point as Boston Logan Airport. To proceed, I need to set the destination to North Station, allowing the trip planner to suggest routes.

Low-level Instruction: Click on the 'To' input field and type 'North Station' as the destination.

Action:
pyautogui.click(x=0.6756, y=0.4)
pyautogui.write(text='North Station')

AGUVIS Collection: Open-source Unified Large Scale GUI Agent Data

salesforce

Stage 1: 1M+ GUI Grounding

Data source	Platform	Instruction	#Trajectory
SeeClick (Cheng et al., 2024)	Website	Augmented	271K
GUIEnv (Chen et al., 2024a)	Website	Augmented	328K
GUIAct (Chen et al., 2024a)	Website	Original	67K
WebUI (Wu et al., 2023)	Website	Augmented	57K
Widget Captioning (Li et al., 2020b)	Mobile	Original	101K
RicoSCA (Li et al., 2020a)	Mobile	Original	173K
UI RefExp (Bai et al., 2021)	Mobile	Original	16K
RICO Icon (Deka et al., 2017)	Mobile	Augmented	16K
OmniACT (Kapoor et al., 2024)	Desktop & Website	Original	7K
Total			1.036M

Stage 2: 35K multi-step trajectories with explicit inner monologue

Data source	Platform	Inner Monologue	Avg. Steps	#Trajectory
MM-Mind2Web (Zheng et al., 2024a)	Website	Generated	7.7	1,009
GUIAct (Chen et al., 2024a)	Website	Generated	6.7	2,482
MiniWoB++ (Zheng et al., 2024b)	Website	Generated	3.6	2,762
AitZ (Zhang et al., 2024b)	Mobile	Original	6.0	1,987
AndroidControl (Li et al., 2024d)	Mobile	Original	5.5	13,594
GUI Odyssey (Lu et al., 2024)	Mobile	Generated	15.3	7,735
AMEX (Chai et al., 2024)	Mobile	Generated	11.9	2,991
AitW (Rawles et al., 2024b)	Mobile	Generated	8.1	2,346
Total				35K

Evaluation: GUI Grounding

Planner	Grounder	Mobile		Desktop		Web		Avg
		Text	Icon/Widget	Text	Icon/Widget	Text	Icon/Widget	
-	GPT-4	22.6	24.5	20.2	11.8	9.2	8.8	16.2
	GPT-4o	20.2	24.9	21.1	23.6	12.2	7.8	18.3
	CogAgent	67.0	24.0	74.2	20.0	70.4	28.6	47.4
	SeeClick	78.0	52.0	72.2	30.0	55.7	32.5	53.4
	Qwen2-VL	75.5	60.7	76.3	54.3	35.2	25.7	55.3
	UGround	82.8	60.3	82.5	63.6	80.4	70.4	73.3
	AGUVIS-G-7B	**88.3**	**78.2**	**88.1**	**70.7**	**85.7**	**74.8**	**81.8**
GPT-4	SeeClick	76.6	55.5	68.0	28.6	40.9	23.3	48.8
	OmniParser	93.9	57.0	91.3	63.6	81.3	51.0	73.0
	UGround	90.1	70.3	87.1	55.7	85.7	64.6	75.6
GPT-4o	SeeClick	81.0	59.8	69.6	33.6	43.9	26.2	52.3
	UGround	93.4	76.9	92.8	67.9	88.7	68.9	81.4
AGUVIS-7B		**95.6**	77.7	93.8	67.1	88.3	75.2	**84.4**
AGUVIS-72B		94.5	**85.2**	**95.4**	**77.9**	**91.3**	**85.9**	**89.2**

Offline Agent Evaluation: Mind2Web

Obs.	Planner	Grounder	Cross-Task			Cross-Website			Cross-Domain		
			Ele.Acc	**Op.F1**	**Step SR**	**Ele.Acc**	**Op.F1**	**Step SR**	**Ele.Acc**	**Op.F1**	**Step SR**
T	GPT-3.5	Choice	19.4	59.2	16.8	14.9	56.5	14.1	25.2	57.9	24.1
	GPT-4	Choice	40.8	63.1	32.3	30.2	61.0	27.0	35.4	61.9	29.7
T + I	GPT-4	Choice	46.4	73.4	40.2	38.0	67.8	32.4	42.4	69.3	36.8
	GPT-4	SoM	29.6	-	20.3	20.1	-	13.9	27.0	-	23.7
I	GPT-4o	SeeClick	32.1	-	-	33.1	-	-	33.5	-	-
	GPT-4V	OmniParser	42.4	87.6	39.4	41.0	84.8	36.5	45.5	85.7	42.0
	GPT-4o	UGround	47.7	-	-	46.0	-	-	46.6	-	-
I	SeeClick-9.6B		28.3	87.0	25.5	21.4	80.6	16.4	23.2	84.8	20.8
	AGUVIS-7B		64.2	89.8	60.4	60.7	88.1	54.6	60.4	**89.2**	56.6
	AGUVIS-72B		**69.5**	**90.8**	**64.0**	**62.6**	**88.6**	**56.5**	**63.5**	88.5	**58.2**

Offline Agent Evaluation: AndroidControl

Obs.	Planner	Grounder	Step Acc. High	Step Acc. Low
Acc. Tree	GPT-4-Turbo	Choice	42.1	55.0
	PaLM 2S*	Choice	58.5	77.5
Image	GPT-4-Turbo	SeeClick	39.4	47.2
	GPT-4-Turbo	UGround	46.2	58.0
	GPT-4o	SeeClick	41.8	52.8
	GPT-4o	UGround	48.4	62.4
Image	AGUVIS-7B		**61.5**	**80.5**
	AGUVIS-72B		**66.4**	**84.4**

Online Agent Evaluation

Inputs	Planner	Grounder	SR	Cost
HTML	GPT-4-Turbo	Choice	21.1	-
	GPT-4o	Choice	22.1	0.142
	Llama-3.1-405B	Choice	24.0	0.174
	Llama-3.1-70B	Choice	20.2	0.031
	GPT-3.5-turbo	Choice	17.3	0.092
Image	GPT-4-Turbo	UGround	23.1	-
	GPT-4o	UGround	19.2	-
	GPT-4o	AGUVIS-7B	**24.0**	**0.106**
Image	AGUVIS-72B		**27.1**	**0.012**

Browser Use (Mind2Web-live)

Input	Planner	Grounder	AW_{SR}	MMW_{SR}
AXTree	GPT-4-Turbo	Choice	30.6	59.7
	Gemini 1.5 Pro	Choice	19.4	57.4
Image + AXTree	GPT-4-Turbo	SoM	25.4	67.7
	Gemini 1.5 Pro	SoM	22.8	40.3
Image	GPT-4-Turbo	UGround	31.0	-
	GPT-4o	UGround	32.8	-
	GPT-4o	AGUVIS-7B	**37.1**	**55.0**
Image	AGUVIS-72B		**26.1**	**66.0**

Mobile Use (AndroidWorld)

Analysis: Impact of Training Stages and Inner Monologue

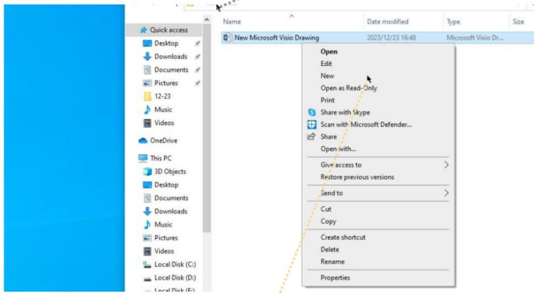
- Both stages 1 and 2 contribute to Aguvis's performance.
- Inner monologue is crucial for both high-level reasoning and low-level action grounding.

Settings	ScreenSpot	Multimodal-Mind2Web			AndroidControl	
		Cross-Task	Cross-Website	Cross-Domain	High-Level	Low-Level
AGUVIS-7B	84.4	60.4	54.6	56.6	61.5	80.5
(a) w/o Stage 2	81.8	50.9	45.2	45.3	58.0	75.6
(b) w/o Stage 1	77.4	59.7	55.3	56.8	58.8	79.8
(c) w/o Stage 1 & 2	55.3	50.9	44.9	47.7	59.1	59.2
(d) w/o Inner Monologue	79.3	55.4	53.7	54.9	60.3	69.1

Reasoning with inner monologue helps solving harder tasks.

Task Instruction: add a new file

pyautogui.click(x=0.3997, y=0.0148) ❌

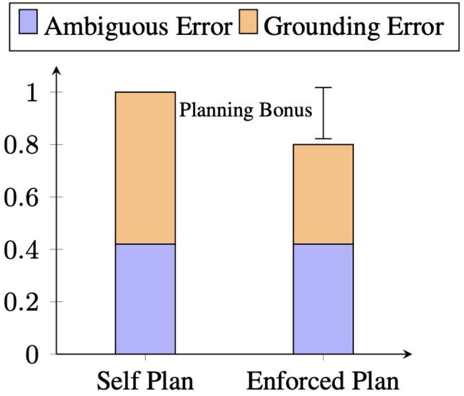


Thought: To add a new file, I need to select the 'New' option from the context menu. This will likely open a dialog or prompt to create a new file, aligning with the goal of adding a new file to the directory.
Action: Click on the 'New' option in the context menu.
pyautogui.click(x=0.6836, y=0.2535) ✅

Task Instruction: send current page

pyautogui.click(x=0.9792, y=0.8046) ❌

Thought: To share the current page, I need to find and click the share icon, which is typically represented by a network or link symbol. This icon is usually located in the browser's toolbar or menu.
Action: Click the share icon in the browser to share the current page.
pyautogui.click(x=0.9792, y=0.876) ✅



Analysis: Cross-Platform Benefits

Despite being trained exclusively on web and mobile trajectory data, our model demonstrates strong generalization to desktop GUI tasks.

Data	**#Traj.**	**Task**	**Website**	**Domain**
Web + Mobile	35k	58.5	55.4	54.8
Web Only	6k	53.1	50.3	52.2
Mind2Web Only	1k	50.9	44.9	47.7

Planner	**Grounding**	**Task SR**
GPT-4o	SoM	4.59
GPT-4o	AGUVIS-7B	14.79
GPT-4o	AGUVIS-72B	17.04
GPT-4o		5.03
GPT-4V		5.26
Gemini-Pro-1.5		5.40
Claude Computer-Use		14.9
OpenAI Operator		**19.7**
AGUVIS-72B		10.26

Takeaways

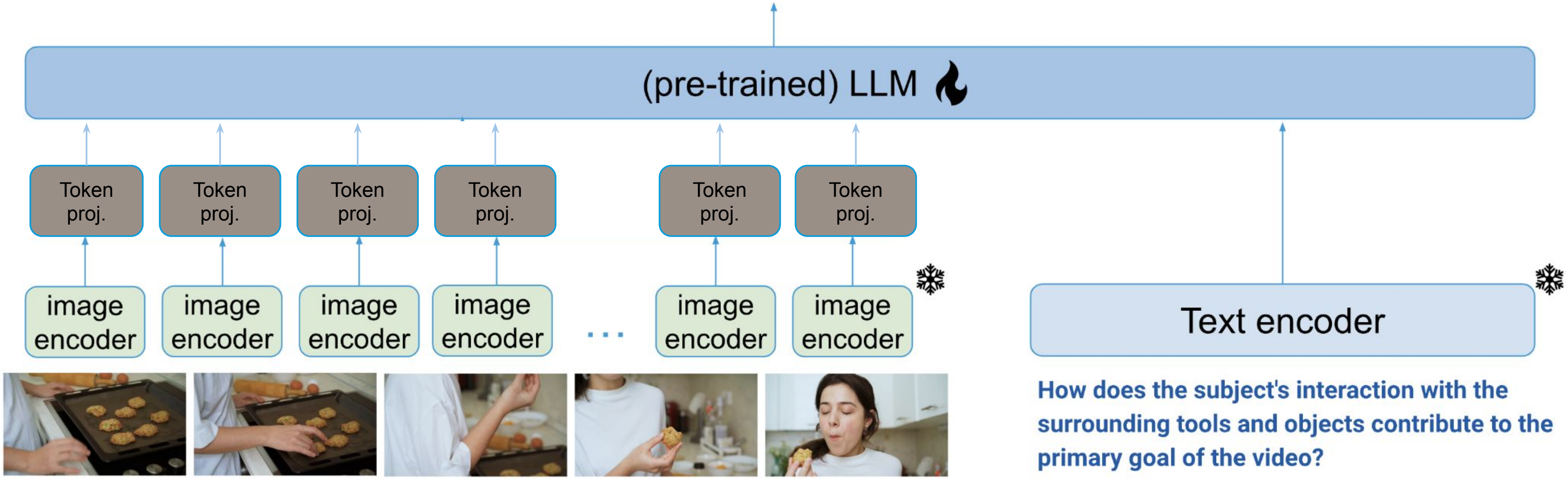
- AGUVIS is a unified framework that enables autonomous GUI agents to operate across different platforms using only visual observations.
- We need to improve both grounding and structured reasoning.

- Next, let's discuss a bit on Video LLM.

Long Video meets Multimodal Agent

Standard approach for Video LMs

- Each frame encoded independently
- Concatenate per-frame token representations



Sequential models

Problem/motivation

We have too many tokens from too many frames

- Long-form videos: 10s of minutes, 1000s of frames -> 100,000s of tokens.

Not only for **videos**, but also for text and multimodal **VLA**

- Capture important details in long videos

Objective

Construct a new SOTA foundation model for long sequential data

- Memory-based models like **Token Turing Machines**

Be efficient!

xGen-MM-Vid (BLIP-3-Video)

Introducing a new efficient video foundation model

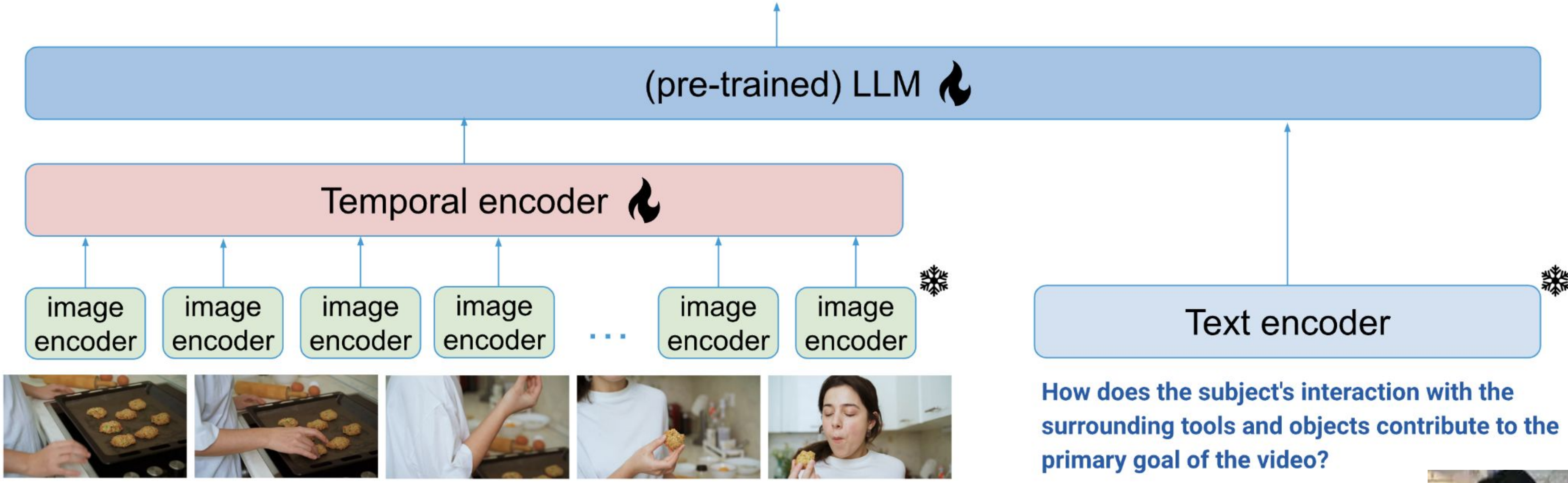
BLIP-3-Video: You Only Need 32 Tokens to Represent a Video Even in VLMs, (Ryoo et.al. 2024)

xGen-MM-Vid (BLIP-3-Video)

Extension from xGen-MM (for images).

“Temporal encoder” abstracts a video into a small # of visual tokens

- 32~128 tokens per video



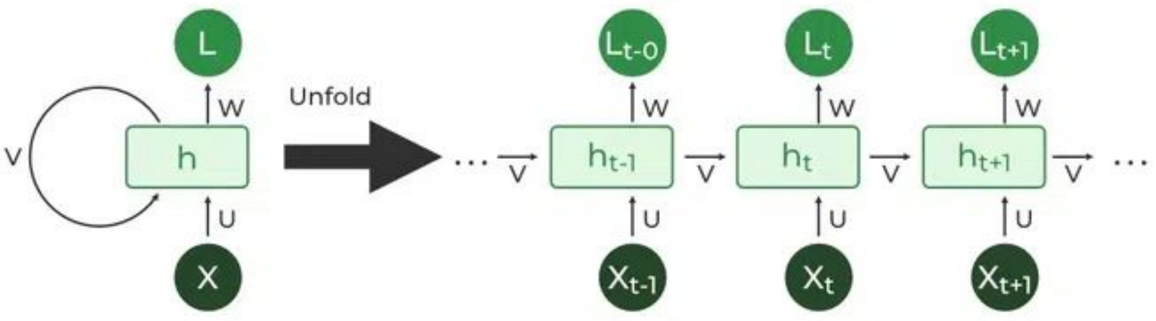
BLIP-3-Video: You Only Need 32 Tokens to Represent a Video Even in VLMs, (Ryoo et.al. 2024)

Sequential models

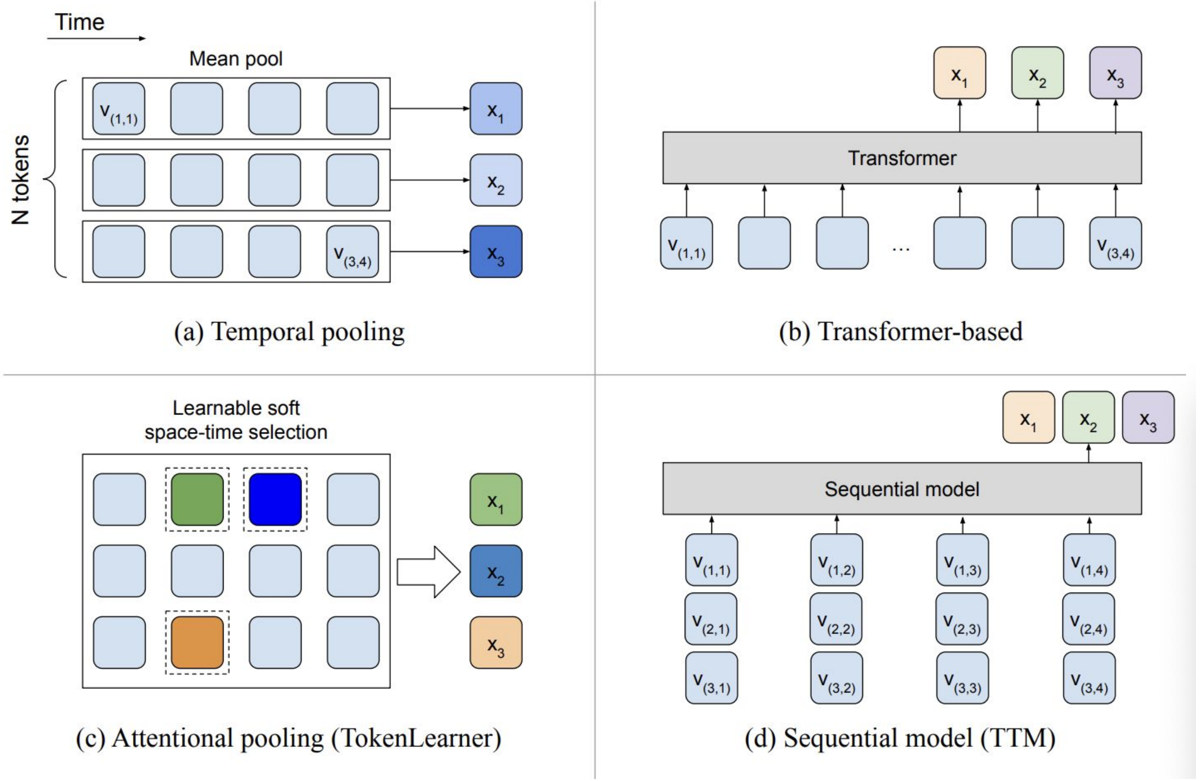
Background

What are sequential models?

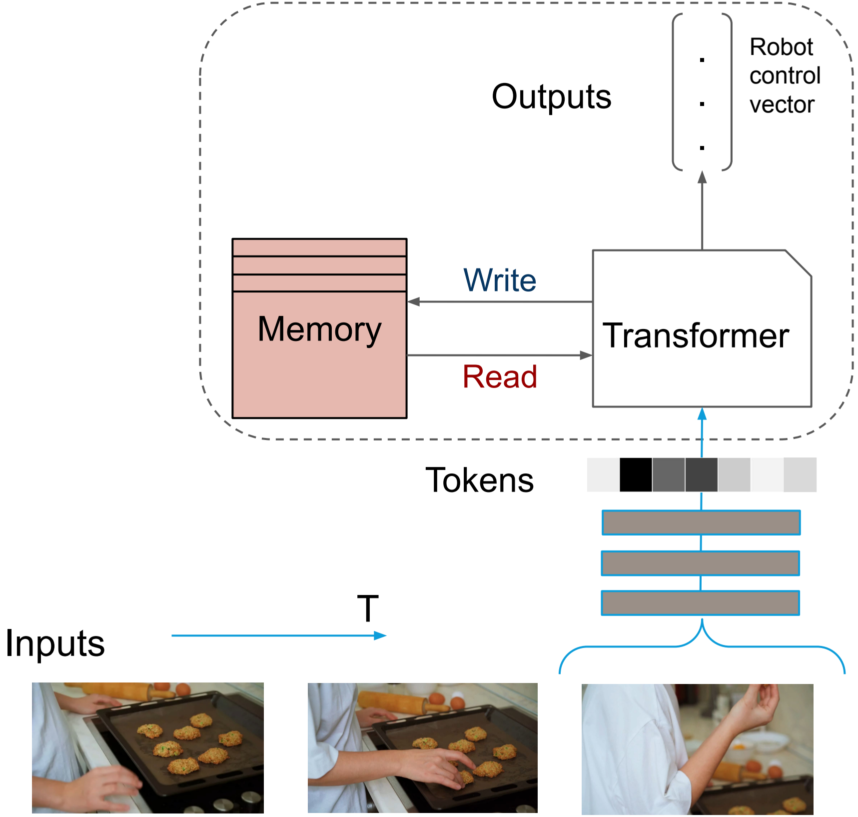
- They take a sequence as an input
- They iterative process per-step input at a time.
- An easy example: LSTM/RNN



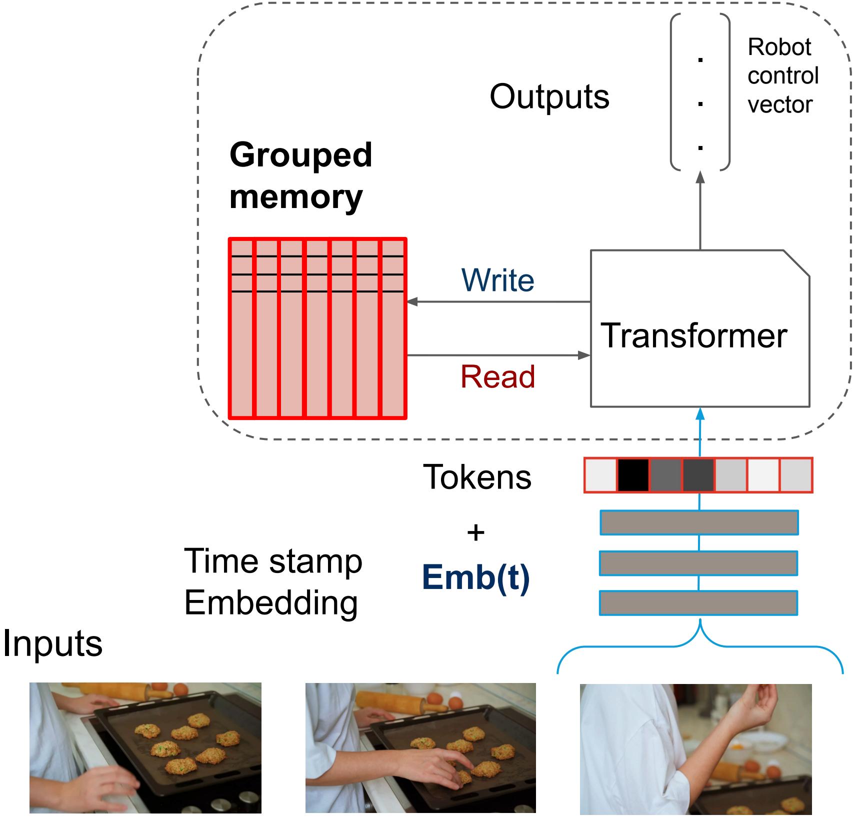
Different types of temporal encoders



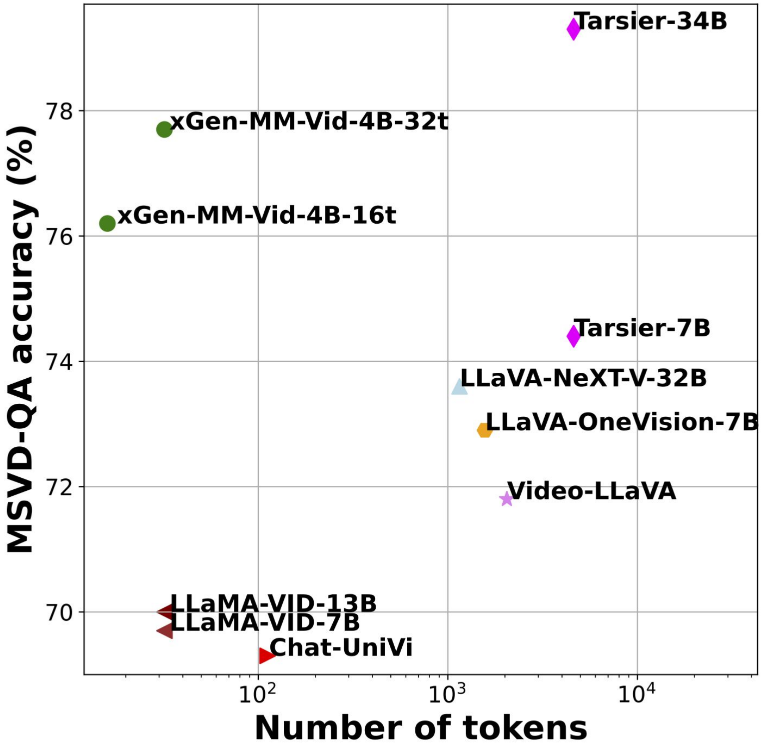
(previous) Token Turing Machine



Our new sequential encoder



xGen-MM-Vid



Compared to other state-of-the-art models, xGen-MM-Vid uses **significantly less** number of visual tokens (32 vs. 4608).

[arXiv paper]
[website]
[tweet]
[podcasts]
[News articles]

Experimental results

Method	Size	#tokens	MSVD-QA	MSRVTT-QA	ActivityNet-QA	TGIF-QA
VideoChat (Li et al., 2023b)	7B	**32**	56.3 / 2.8	45.0 / 2.5	- / 2.2	34.4 / 2.3
Video-LLaMA (Zhang et al., 2023)	7B	**32**	51.6 / 2.5	29.6 / 1.8	12.4 / 1.1	- / -
Video-ChatGPT (Maaz et al., 2024)	7B	264+	64.9 / 3.3	49.3 / 2.8	34.2 / 2.8	51.4 / 3.0
Chat-UniVi (Jin et al., 2024)	7B	112	69.3 / 3.7	55.0 / 3.1	46.1 / 3.3	69.0 / 3.8
LLaMA-VID (Li et al., 2024c)	7B	**32**	69.7 / 3.7	57.7 / 3.2	47.4 / 3.3	-
LLaMA-VID (Li et al., 2024c)	13B	**32**	70.0 / 3.7	58.9 / 3.3	47.5 / 3.3	-
Video-LLaVA (Lin et al., 2023)	7B	2048	71.8 / 3.9	59.2 / 3.5	45.3 / 3.3	70.0 / 4.0
MiniGPT4-Video (Ataallah et al., 2024)	7B	2880+	73.9 / 4.1	59.7 / 3.3	46.3 / 3.4	72.2 / 4.1
PLLaVA (Xu et al., 2024a)	7B	576+	76.6 / 4.1	62.0 / 3.5	56.3 / 3.5	77.5 / 4.1
SlowFast-LLaVA Xu et al. (2024b)	7B	3680	79.1 / 4.1	65.8 / 3.6	56.3 / 3.4	78.7 / 4.2
LLaVA-Hound-DPO Zhang et al. (2024b)	7B	2048	80.7 / 4.1	70.2 / 3.7	- / -	61.4 / 3.5
LLaVA-OneVision* (Wang et al., 2024a)	7B	1568	72.9 / 3.9	57.8 / 3.4	55.3 / 3.6	41.1 / 3.1
Tarsier (Wang et al., 2024a)	7B	4608+	77.0 / 4.1	62.0 / 3.5	59.5 / 3.6	79.2 / 4.2
Tarsier * (Wang et al., 2024a)	7B	4608	74.4 / 4.0	59.1 / 3.4	54.3 / 3.5	- / -
PLLaVA (Xu et al., 2024a)	34B	576+	79.9 / 4.2	68.7 / 3.8	60.9 / 3.7	80.6 / 4.3
LLaVA-NeXT-Video* (Li et al., 2024b)	32B	1152	73.6 / 4.0	56.8 / 3.4	58.4 / 3.6	73.5 / 4.1
Tarsier (Wang et al., 2024a)	34B	4608+	80.3 / 4.2	66.4 / 3.7	61.6 / 3.7	82.5 / 4.4
Tarsier * (Wang et al., 2024a)	34B	4608+	79.3 / 4.1	62.2 / 3.5	61.5 / 3.7	- / -
BLIP-3-Video	**4B**	**32**	77.1 / 4.2	60.0 / 3.6	55.7 / 3.5	77.1 / 4.3
BLIP-3-Video	**4B**	128	77.3 / 4.2	59.7 / 3.6	56.7 / 3.6	77.1 / 4.3

Multiple choice question - experiments

Method	Size	#tokens	NExT-QA
LangRepo (Kahatapitiya et al., 2024)	7B	3136+	54.6
LangRepo (Kahatapitiya et al., 2024)	12B	3136+	60.9
Tarsier (Wang et al., 2024a)	7B	4608+	71.6
LLoVi (Zhang et al., 2024a)	157B	1000s	67.7
IG-VLM (Kim et al., 2024)	34B	1536+	70.9
VideoAgent (Wang et al., 2024b)	GPT-4	2091+	71.3
VideoTree (Wang et al., 2024c)	GPT-4	3978+	73.5
Tarsier (Wang et al., 2024a)	34B	4608+	79.2
BLIP-3-Video	**4B**	**32**	76.4
BLIP-3-Video	**4B**	128	77.1

Ablation: Sequential model

Temporal encoder	MSVD-QA	TGIF-QA	ActivityNet-QA	NExT-QA
Original TTM	76.42 / 4.15	75.80 / 4.26	54.45 / 3.48	75.42
TTM + time-stamp	76.43 / 4.16	76.44 / 4.29	56.15 / 3.53	75.96
TTM + grouping	76.99 / 4.17	77.05 / 4.30	55.92 / 3.54	76.46
Ours (time-stamp + grouping)	77.29 / 4.18	77.10 / 4.31	56.66 / 3.56	77.07

Ablation: Number of frames and tokens

Scaling our model (to digest more frames) enable better results

# frames	# tokens	NExT-QA	ActivityNet-QA
8 frames	32 tokens	76.4	55.7 / 3.5
8 frames	128 tokens	77.1	56.7 / 3.6
16 frames	32 tokens	76.7	55.9 / 3.5
16 frames	128 tokens	77.6	57.3 / 3.6

Video captioning experiments - examples

Question: Please provide a dense description of the video, covering the main subject, movements, style, backgrounds, and cameras.

GT: A step-by-step guide on assembling a wooden bed frame, featuring a person methodically working through the construction process. Initially, the person is seen preparing the wooden planks and arranging the necessary tools. As the video progresses, the individual meticulously attaches and secures the slats to the frame, ensuring stability and precision. The final scenes depict the person tightening the last screws and reviewing the completed structure, demonstrating a sense of accomplishment and attention to detail.

Tarsier-7b: The video starts with a person assembling a piece of furniture. The scene then transitions to a website showcasing different types of beds, including a bed with a canopied frame and a minimalist bed with a white mattress. The video concludes with social media and website icons for 'dactylomoovis' and 'dactylomoveis'.

LLaVA-OneVision-7b: The video begins with a person in a pink shirt with the number 18 on it, working on assembling a wooden bed frame. The individual is seen using a black brush to apply paint or stain to the wooden parts of the bed frame. The text 'Utilize espacador para colar os estrados' (Use an spacer to glue the slats) appears at the bottom of the screen, providing instructions for the assembly process. The scene transitions to a top-down view of the person continuing to work on the bed frame, with the text 'Parafuse os estrados nas barras' (Screw the slats into the bars) appearing at the bottom, offering further guidance. The background remains consistent throughout, showing a white wall and some tools on the floor. The video then shifts to a promotional website for Dactylo.com.br, featuring images of different bed frames and a brief description of the company's mission to create high-quality furniture using sustainable materials. The website also includes social media icons for Facebook, Twitter, Google+, and YouTube, along with the Dactylo logo and the tagline 'Movimento de Madeira' (Wood Movement). The video concludes with a promotional image for Dactylo.com.br, showcasing a white bed frame with the text 'COMO MANTER A BICAMA MISSION' (How to maintain the Bicama Mission), emphasizing the importance of proper care for the furniture.

BLIP-3-Video: A step-by-step process of assembling a piece of furniture, specifically a bed frame. The person involved is seen working on a wooden bed frame, using various tools and components to construct the bed. The video likely serves as a tutorial, providing viewers with a visual guide on how to assemble the bed frame themselves. The person's actions are deliberate and precise, indicating a level of expertise or familiarity with the task.

Blip-3-video uses online memory. How about offline memory? RAG?

Generative Frame Sampler for Long Video Understanding

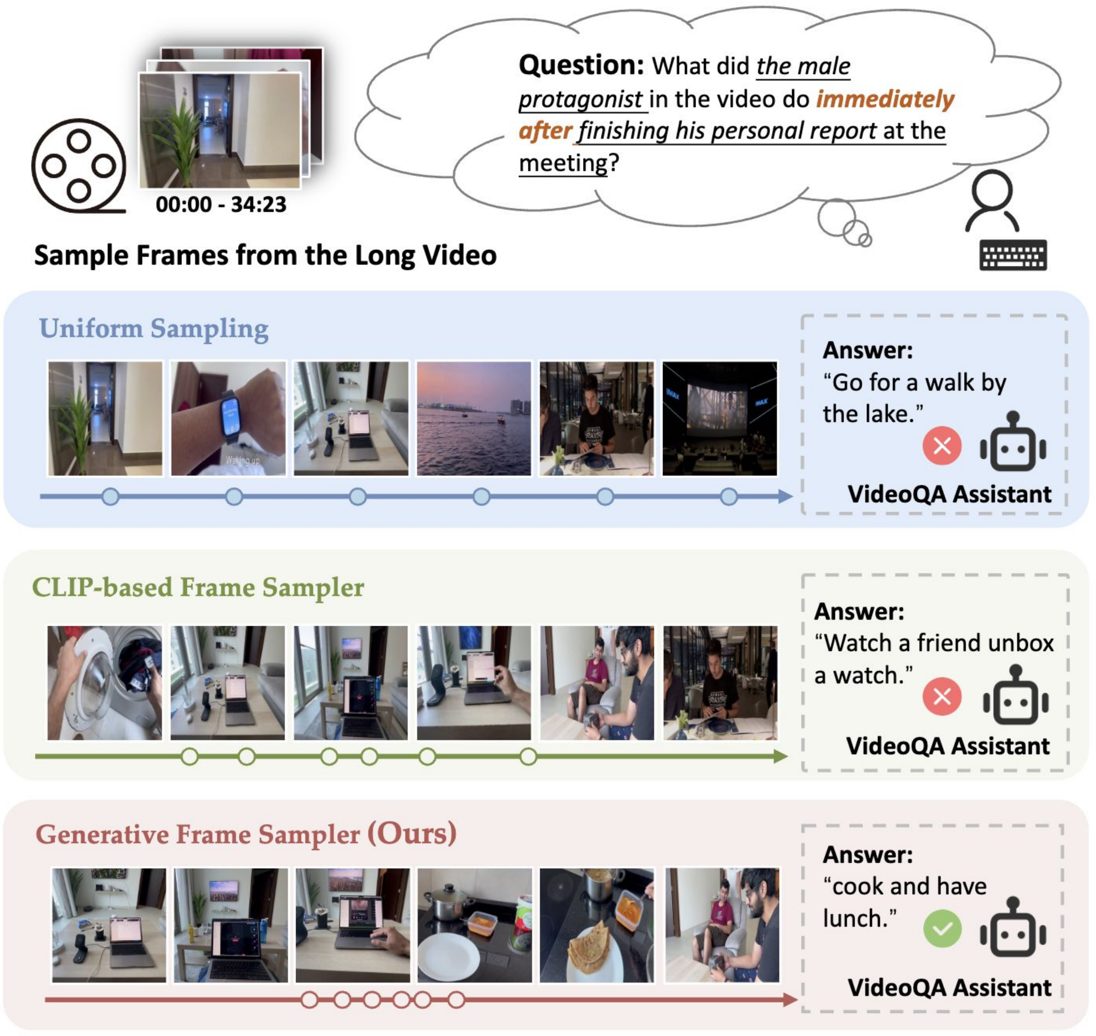
- Understanding long videos containing thousands of frames poses substantial challenge and computational burden to VideoLLMs
- How to efficiently sample representative frames from the original video sequence?

RAG: CLIP-based sampling

- cannot capture temporal relationships between frames

- limited language understanding abilities

- naive cosine similarity cannot achieve multi-hop reasoning

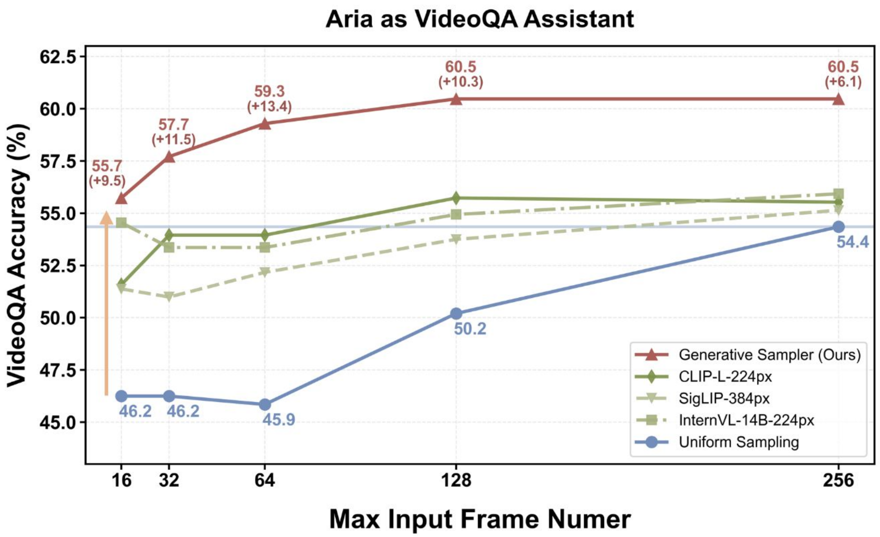


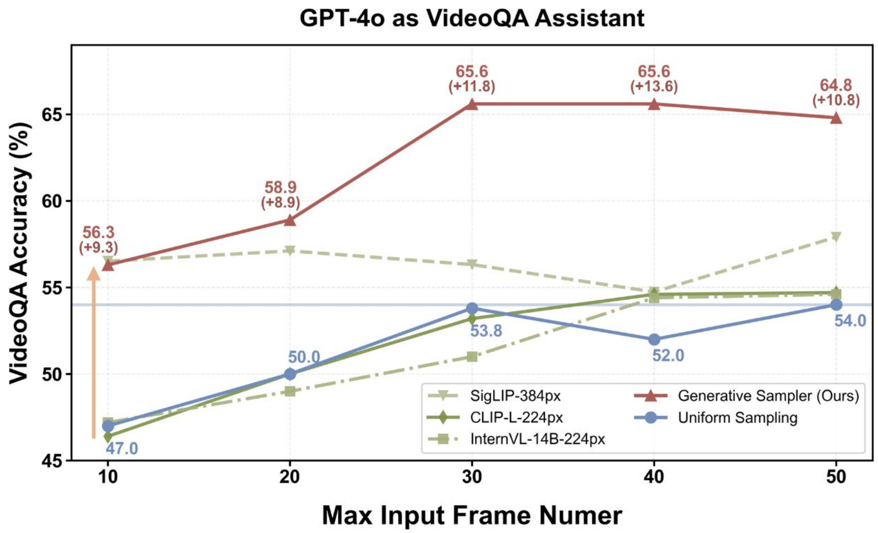
GenS-Video-150K Training Dataset

- *(video, user instruction, relevant frames)* samples that enable the GenS model to identify salient frames for user instructions
- 150K videos with an average duration of 647.5 seconds.
- Among these frames, 20% on average are annotated as relevant with fine-grained confidence scores, providing dense supervision.

Generative Frame Sampler (GenS)

- Built upon an advanced long-context VideoLLM
- Predict relevant frame spans with confidence scores as a natural language generation task *{“Frame Nstart-Nend: relevance score”, …}*
- Significant improvement on long video understanding tasks





Train and Inference

- Fine-tuning based on Aria: MoE with 3.9B activated parameters, SoTA video understanding capabilities
- Output represented as JSON-based format for both discrete frame annotations (e.g., *{“frame number”: relevance score}*) and continuous temporal spans (e.g., *{“start frame - end frame”: relevance score}*)
- Trained on GenS-Video-150K + E.T. Instruct dataset (event localization)

- Inference: sample frames from the input video at 1 FPS, inference within each 256-frame interval using a sliding window approach

GenS significantly improves Long Video QA

VideoQA Model	Size	Frames	LongVideoBench$_{val}$ (avg 12min)		MLVU$_{val}$ (avg 12min)	
			Full	*V-Centric*	*Full*	*V-Centric*
Proprietary LMMs						
GPT-4o	-	256/0.5fps	66.7	-	64.6	-
Gemini-1.5-Pro	-	256/256	64.0	-	-	-
Open-source Video LLMs						
LLaVA-Video	7B	64/64	58.9	50.0	70.4	66.9
LLaVA-Video w/ GenS	7B	54/50	63.3 (+4.4)	56.7 (+6.7)	73.4 (+3.0)	70.6 (+3.7)
Qwen2-VL	7B	64/64	56.0	45.9	64.7	62.3
Qwen2-VL w/ GenS	7B	54/50	58.7 (+2.7)	49.2 (+3.3)	66.9 (+2.2)	64.8 (+2.5)
Aria	25B (3.9B activated)	256/256	62.7	54.4	69.5	62.1
Aria w/ GenS	25B (3.9B activated)	54/95	66.1 (+3.4)	**59.3** (+4.9)	72.6 (+3.1)	67.5 (+5.4)
VILA-v1.5	40B	14/14	57.4	47.0	57.8	52.5
VILA-v1.5 w/ GenS	40B	14/14	59.6 (+2.2)	50.2 (+3.2)	63.5 (+5.7)	58.3 (+5.8)
LLaVA-Video	72B	64/64	62.5	51.6	74.3	72.5
LLaVA-Video w/ GenS	72B	54/50	**66.8** (+4.3)	58.9 (+7.3)	**77.0** (+2.7)	**74.1** (+1.6)

Table 1: Performance on LongVideoBench (Wu et al., 2024a) and MLVU (Zhou et al., 2024) benchmarks using multiple-choice accuracy metrics. *V-Centric* denotes a vision-centric subset containing questions that explicitly require video understanding rather than language-only reasoning, while filtering short videos. Frames *N/M* indicates input N frames for LongVideoBench and M frames for MLVU separately. Using GenS, we select the K most relevant frames (K <= max frame number of VideoQA models) and report the average number of input frames.

GenS is also SoTA on temporal grounding

Grounding Model	Charades-STA R1@0.3	R1@0.5	R1@0.7	mIoU
Temporal Grounding VideoLLMs (7B size)				
VTimeLLM	51.0	27.5	11.4	31.2
HawkEye	50.6	31.4	14.5	33.7
TimeChat[CVPR 2024]	-	32.2	13.4	30.6
TimeSuite[ICLR 2025]	69.9	48.7	24.0	-
General VideoLLMs				
GPT-4o	55.0	32.0	11.5	35.4
VideoChat2-7B	9.6	3.4	1.4	-
Qwen2-VL-7B	8.7	5.4	2.4	7.9
LongVA-7B-DPO	22.6	10.1	2.2	14.6
LLaVA-OneVison-7B	31.2	13.5	5.2	-
Aria	39.0	18.6	6.6	26.7
GenS	62.9	38.7	15.2	38.0
GenS w/o E.T.Instruct-41K$_{agg.}$	51.1	28.2	10.4	33.2

Table 4: Results on the Charades-STA (Gao et al., 2017) temporal grounding benchmark.

Summary

- OSWorld (Environment)
- Agenttreck (Data Synthesis)
- TACO (Data Synthesis)
- Aguvis (Grounding & Reasoning)
- Blip-3-Video (Online Memory)
- GenS (Offline Memory)

Thank you!