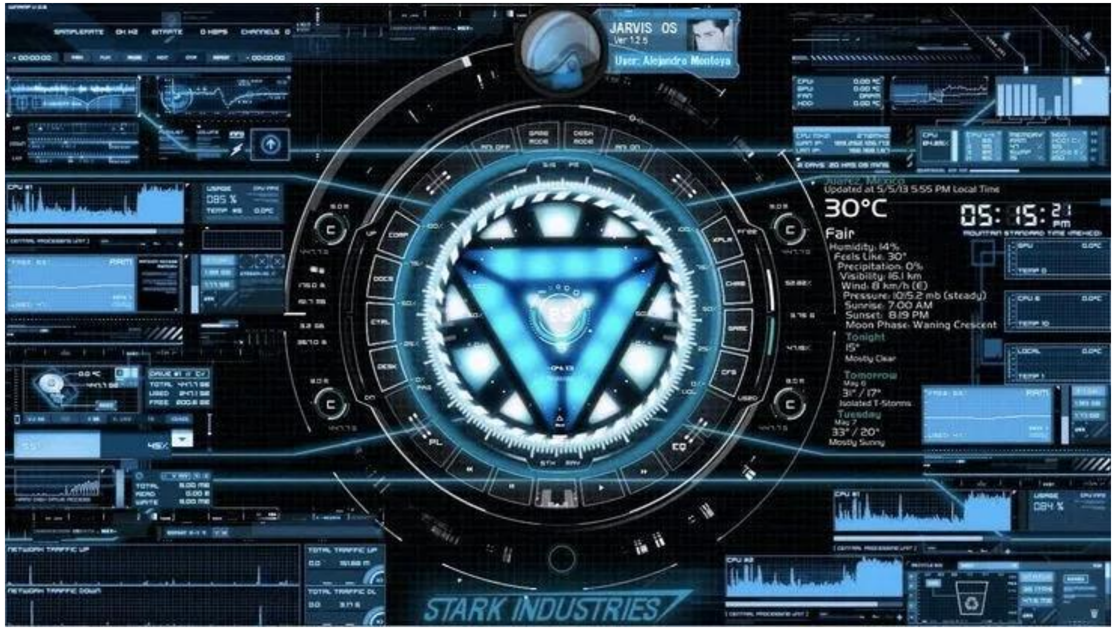
上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

SCHOOL OF COMPUTING & DATA SCIENCE
The University of Hong Kong

Building GUI Agent Data with OS-Genesis

Qiushi Sun
qiushisun.github.io
@qiushi_sun

Computer Use Agents



The Feasibility of Jarvis AI from Marvel in Real Life

[1] OS-Copilot: Towards Generalist Computer Agents with Self-Improvement

Computer Use Agents

Once out of reach, but we are turning it into reality.

Computer Use Agents

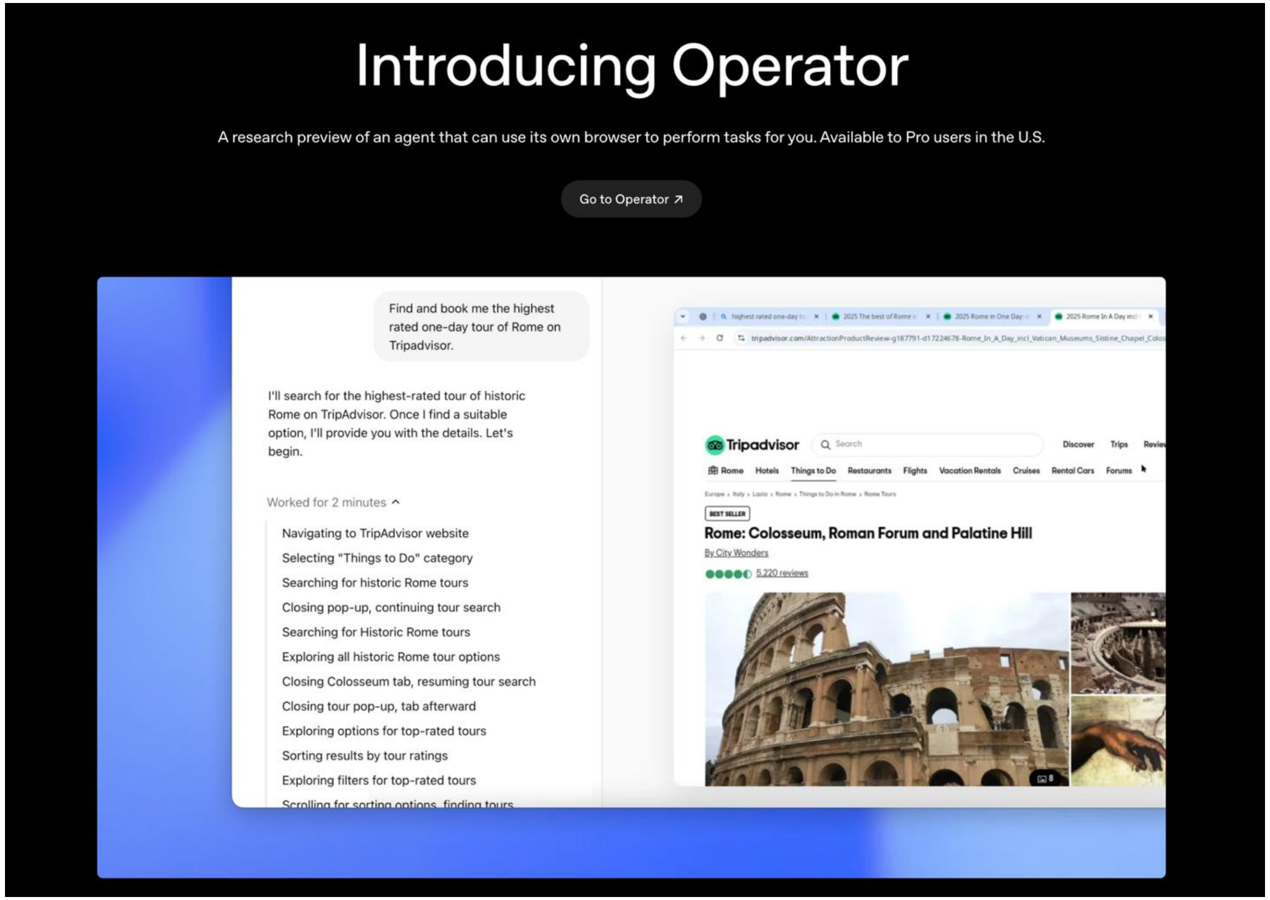
Both academia and industry are building computer use agents

Claude Computer Use

[2] *Introducing computer use, a new Claude 3.5 Sonnet, and Claude 3.5 Haiku, 22 Oct 2024*

Computer Use Agents

Both academia and industry are building computer use agents



OpenAI Operator

[3] *Introducing Operator: A research preview of an agent that can use its own browser to perform tasks for you., Jan 23, 2025*

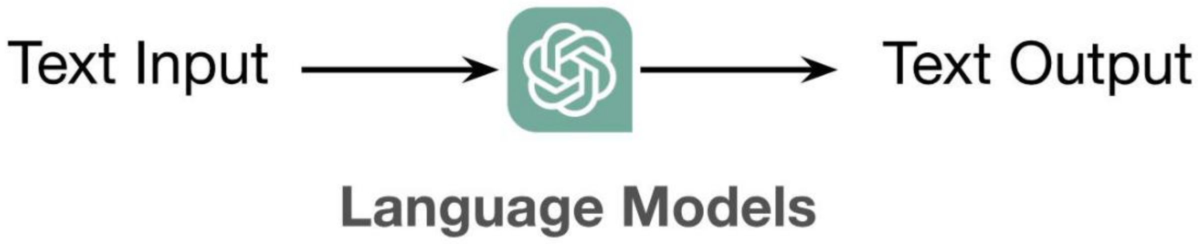
Computer Use Agents

They are quite promising for achieving Digital Automation.

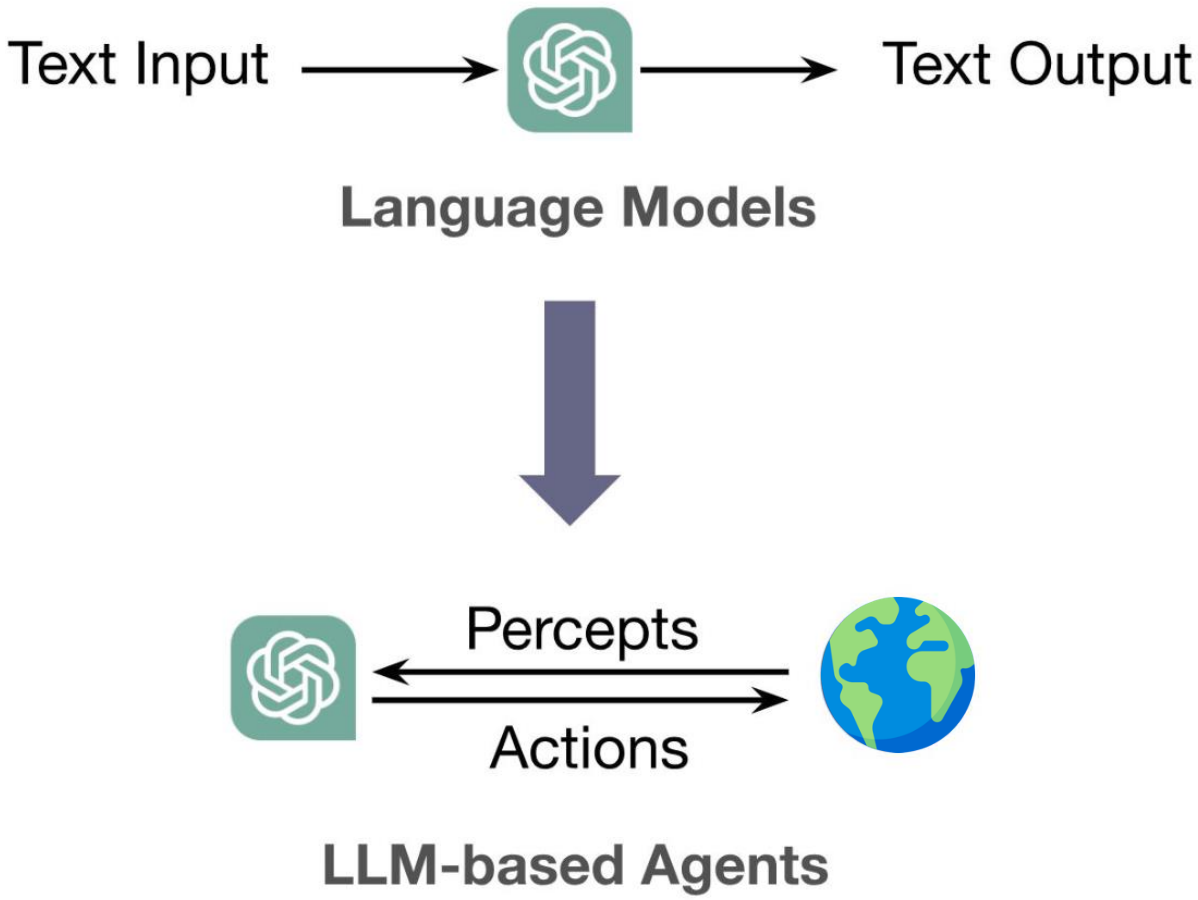
Can we transform a (V)LM into such computer / GUI agents?

Of course! But it is a non-trivial job!

Recap: Language Agents



Recap: Language Agents



But this is not enough for Computer Use / GUI Agents.

Computer Use Agents

Agents are promising, but building powerful agents is challenging.

1. Agents need to follow human instructions.
2. Agents need to perform planning and action.
3. Agents need to perceive envs. and the applications they are interacting with.

Best Way to build Computer Use Agents

Behavioral Cloning / Imitation Learning.

Sounds good, but where is our data?

Data Scarcity

Data curation is much more expensive than you think.

Take Scale AI as an example.

Not to mention scenario/domain - specific data.

[4] *https://x.com/alexandr_wang*

Data Scarcity

How about having the machine collect data?

1. Pre-defined tasks are required, but they may not align with the environment.
2. Limited diversity and a poor success rate.😢

Data Scarcity

So, our goals are as follows:

1. Eliminate human involvement.
2. Obtain high-quality Trajectory data.
3. Diversity and Scalability.

OS-Genesis Automating GUI Agent Trajectory Construction via Reverse Task Synthesis

Qiushi Sun*, Kanzhi Cheng*, Zichen Ding*, Chuanyang Jin*, Yian Wang
Fangzhi Xu, Zhenyu Wu, Liheng Chen, Chengyou Jia, Zhoumianze Liu
Ben Kao, Guohao Li, Junxian He, Yu Qiao, Zhiyong Wu

GUI Trajectory Data

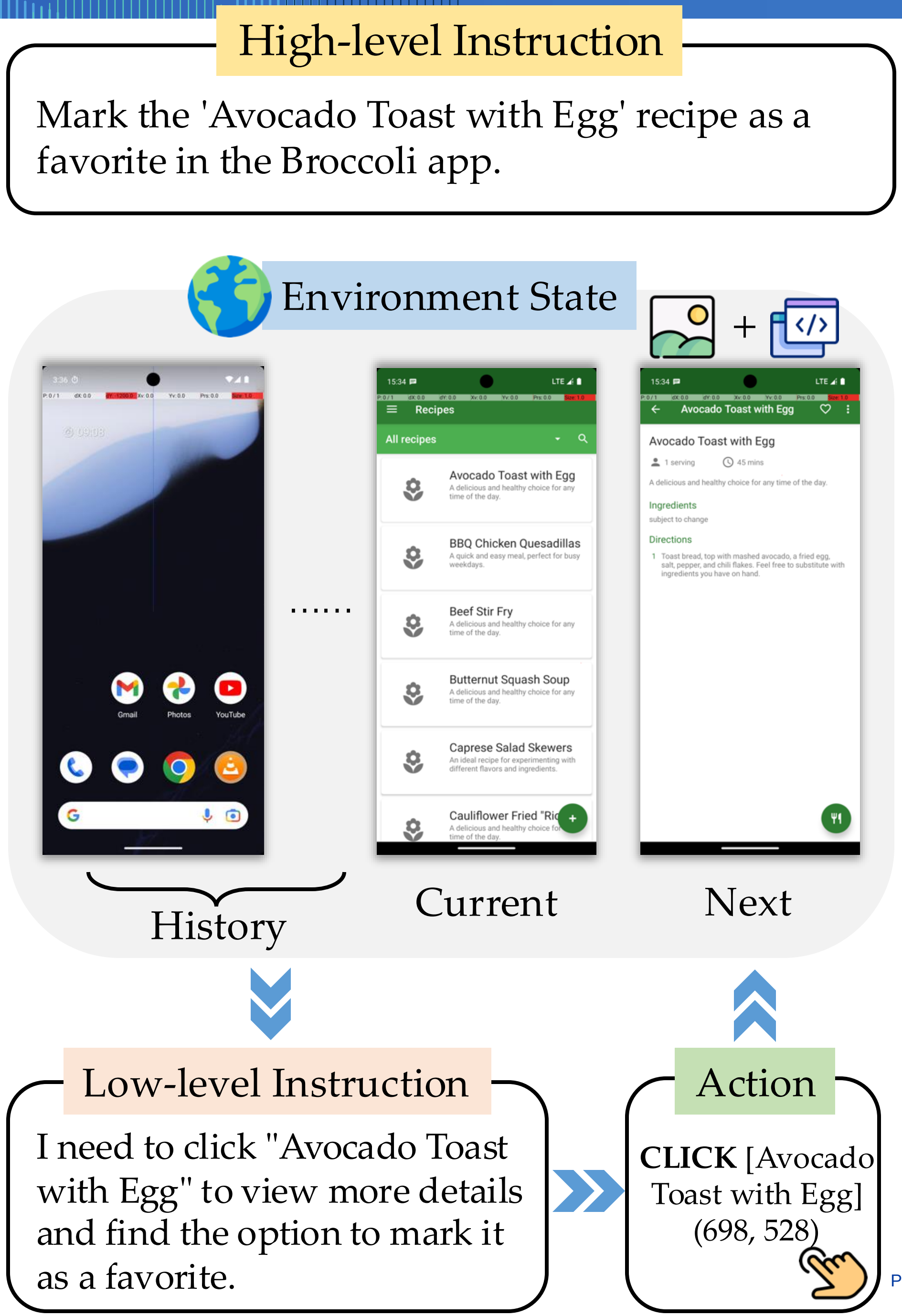
The best data format of GUI agents

1. A high-level instruction that defines the overall goal the agent aims to accomplish

2. A series of low-level instructions that each describe specific steps required

3. Actions (e.g., `CLICK`, `TYPE`)

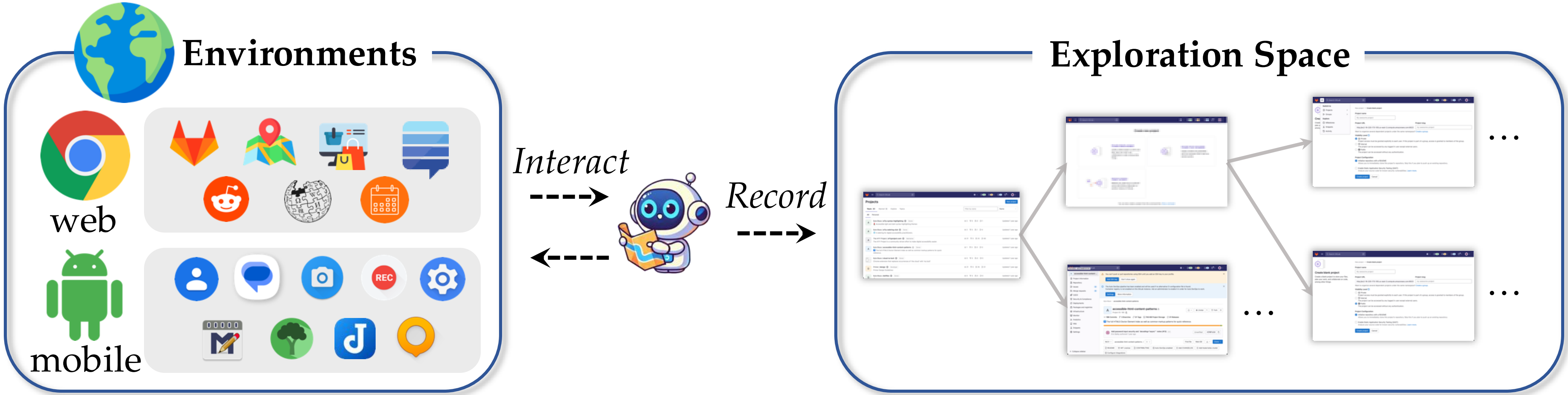
4. States, which include visual representations like screenshots and textual representations such as a11ytree



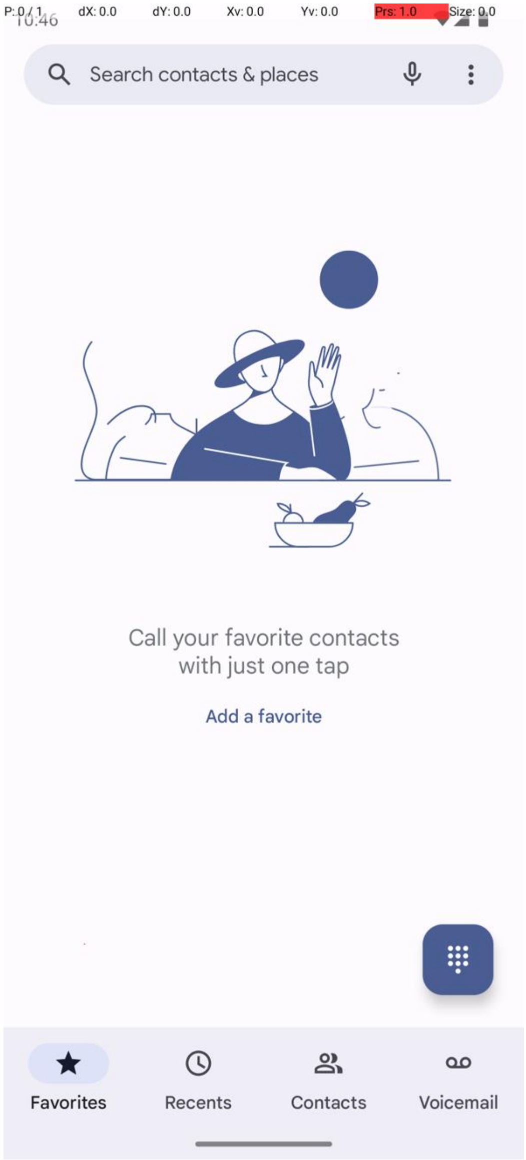
Reverse Task Synthesis

Interaction-Driven Functional Discovery is a rule-based process that explores dynamic GUI environments by interacting with UI elements. It uncovers functionalities through interaction triples

We collect: <Screen1, action, Screen2>

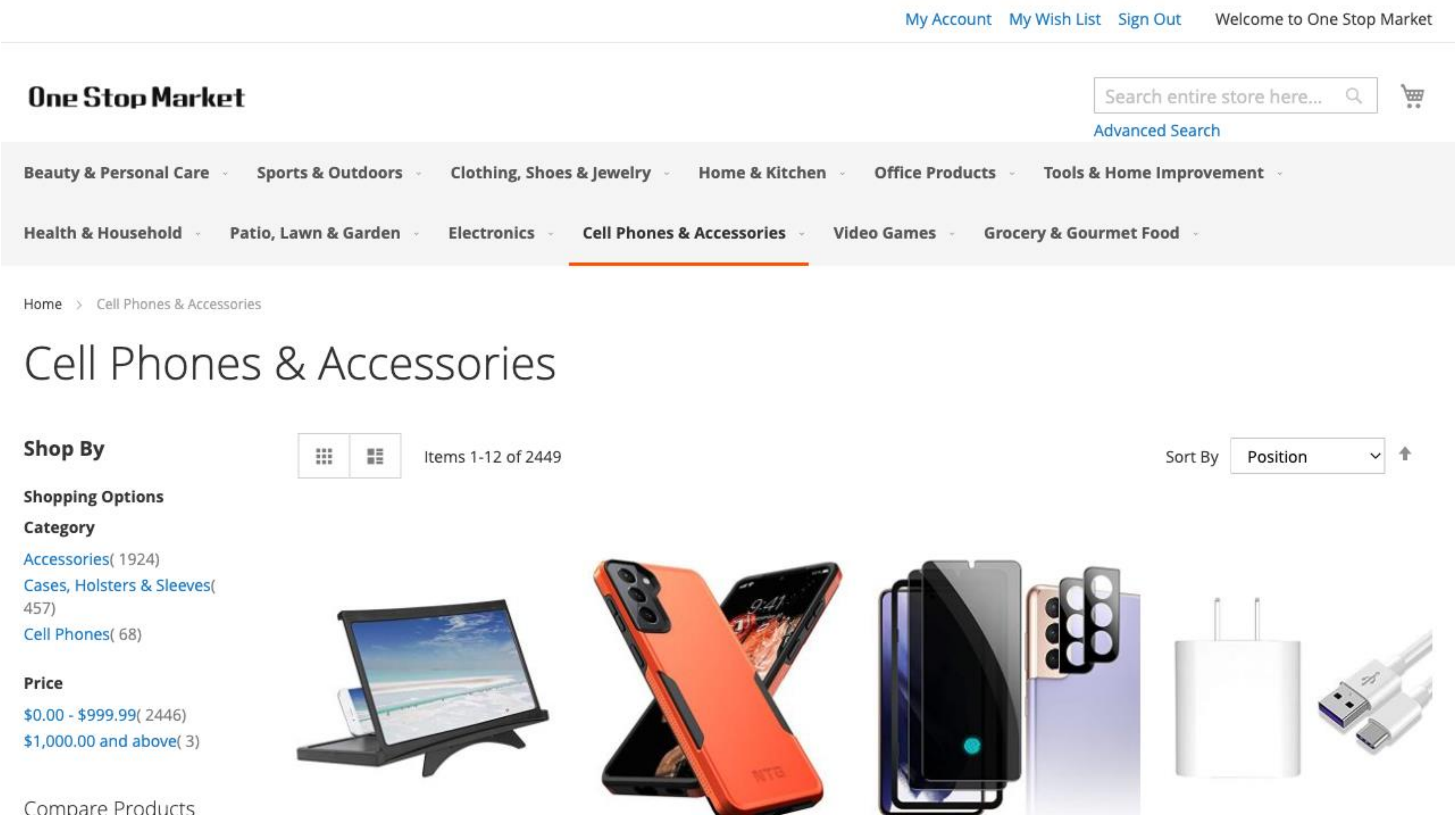


Dynamic Environments



[5] AndroidWorld: A Dynamic Benchmarking Environment for Autonomous Agents, ICLR 2025

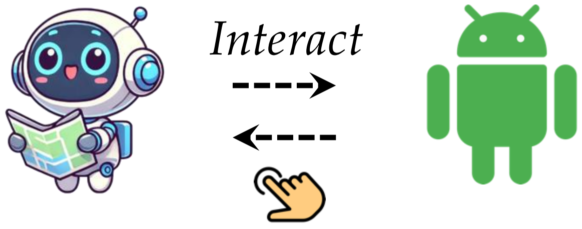
Dynamic Environments



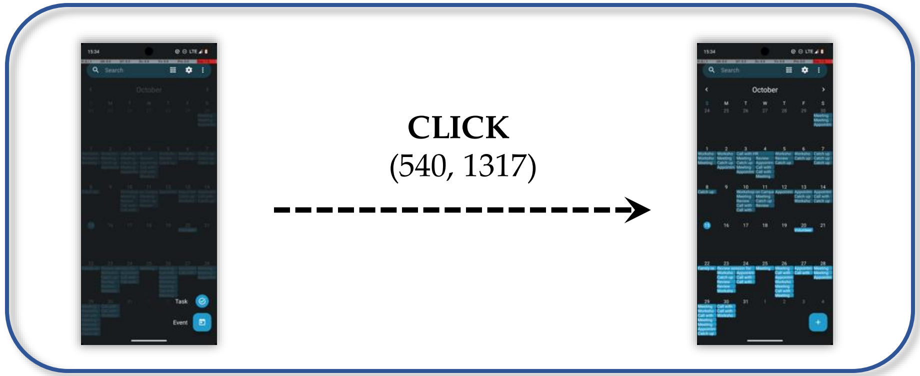
[6] *WebArena: A Realistic Web Environment for Building Autonomous Agents, ICLR 2024*

Reverse Task Synthesis

Retroactively interpreting changes in the GUI environment caused by actions.

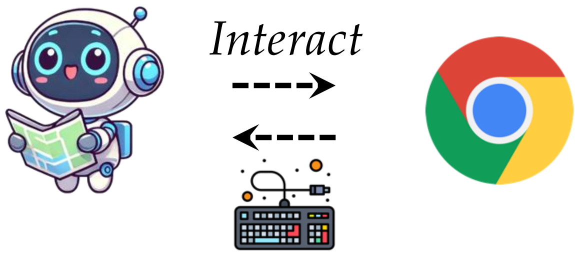


Screenshots & Actions

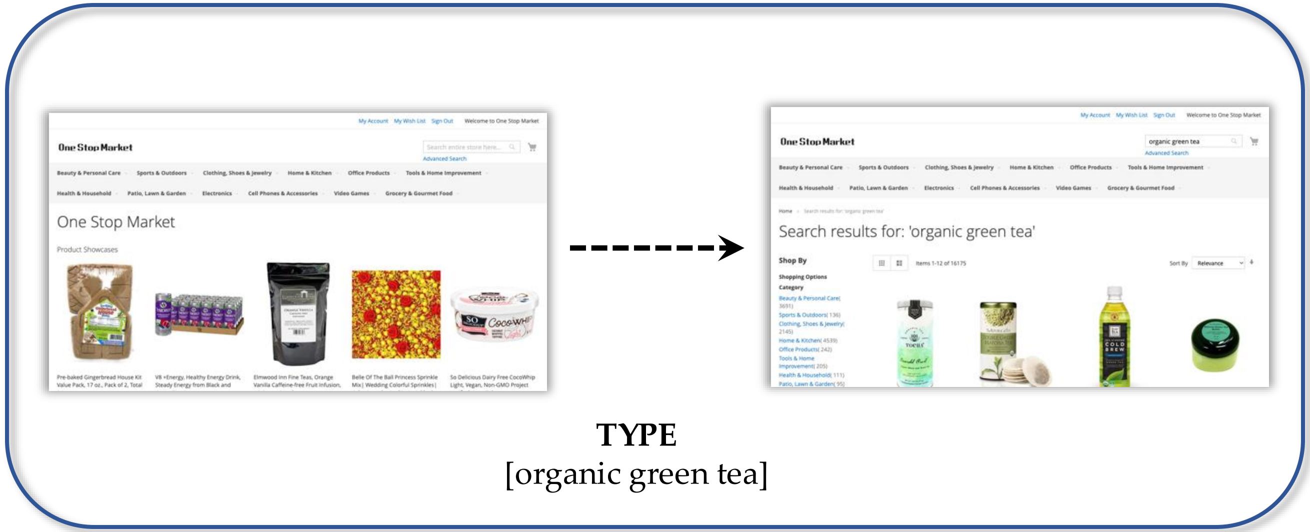


Reverse Task Synthesis

Retroactively interpreting changes in the GUI environment caused by actions.

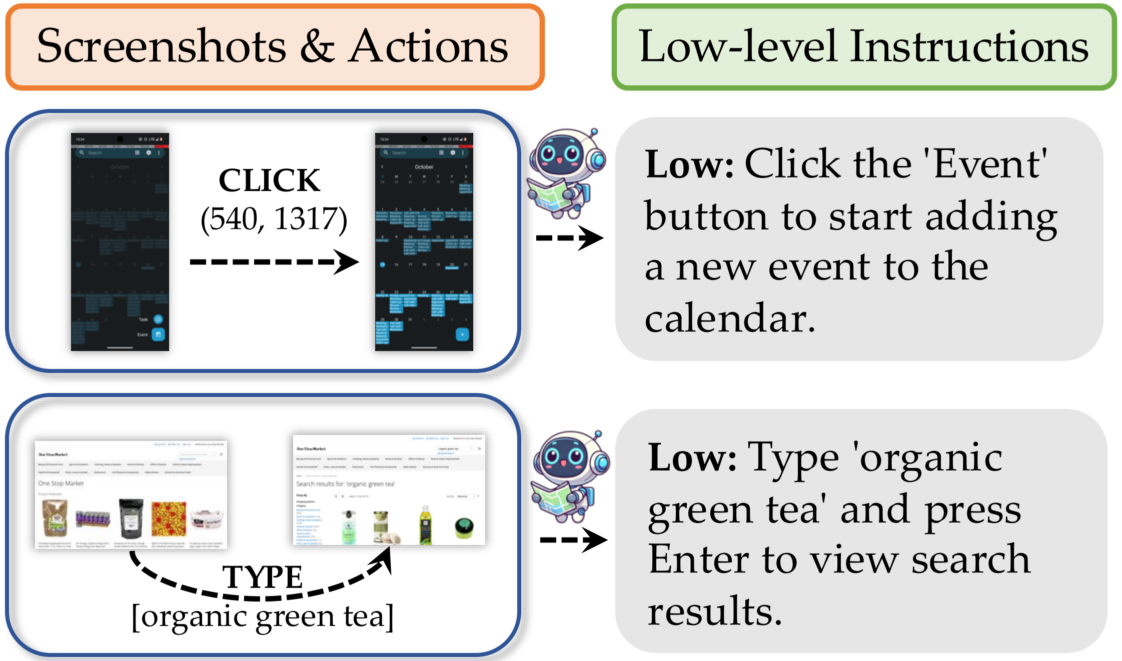


Screenshots & Actions



Reverse Task Synthesis

Retroactively interpreting changes in the GUI environment caused by actions, this process generates executable low-level instructions



The data we synthesized:

1. Grounded

2. Actionable

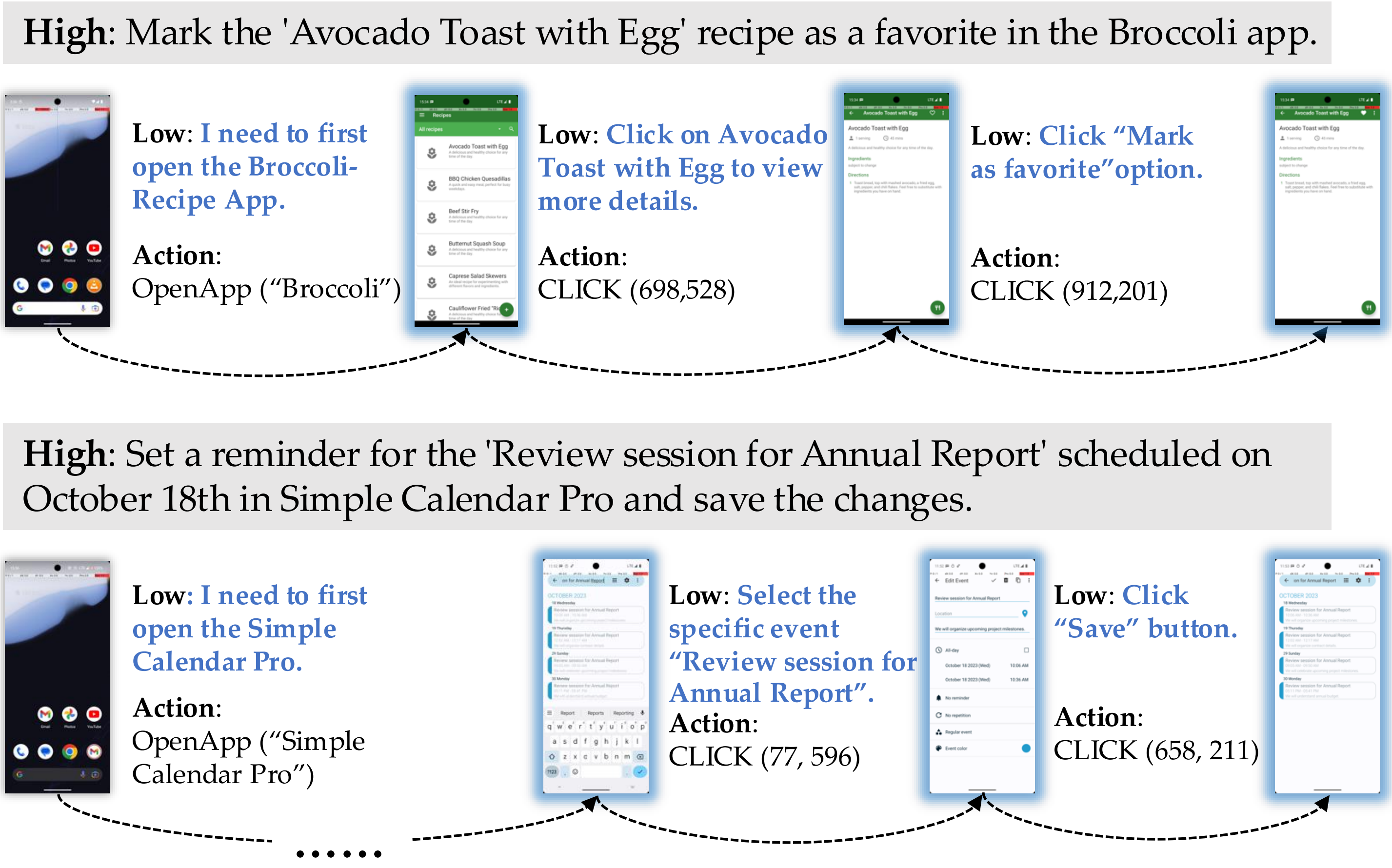
Reverse Task Synthesis

Retroactively interpreting changes in the GUI environment caused by actions, this process generates executable low-level instructions, which are then transformed into broader, goal-oriented high-level tasks

Screenshots & Actions	Low-level Instructions	High-level Instruction
	**Low:** Click the 'Event' button to start adding a new event to the calendar.	**High**: In Simple Calendar Pro, create a new event titled 'Team Meeting' scheduled for October 15, 2023, at 10:00 AM. Save the event after filling in the details.
	**Low:** Type 'organic green tea' and press Enter to view search results.	**High**: Search for 'organic green tea' and filter the results to show only products under the 'Health & Household' category, sorted by price from lowest to highest.

Reverse Task Synthesis

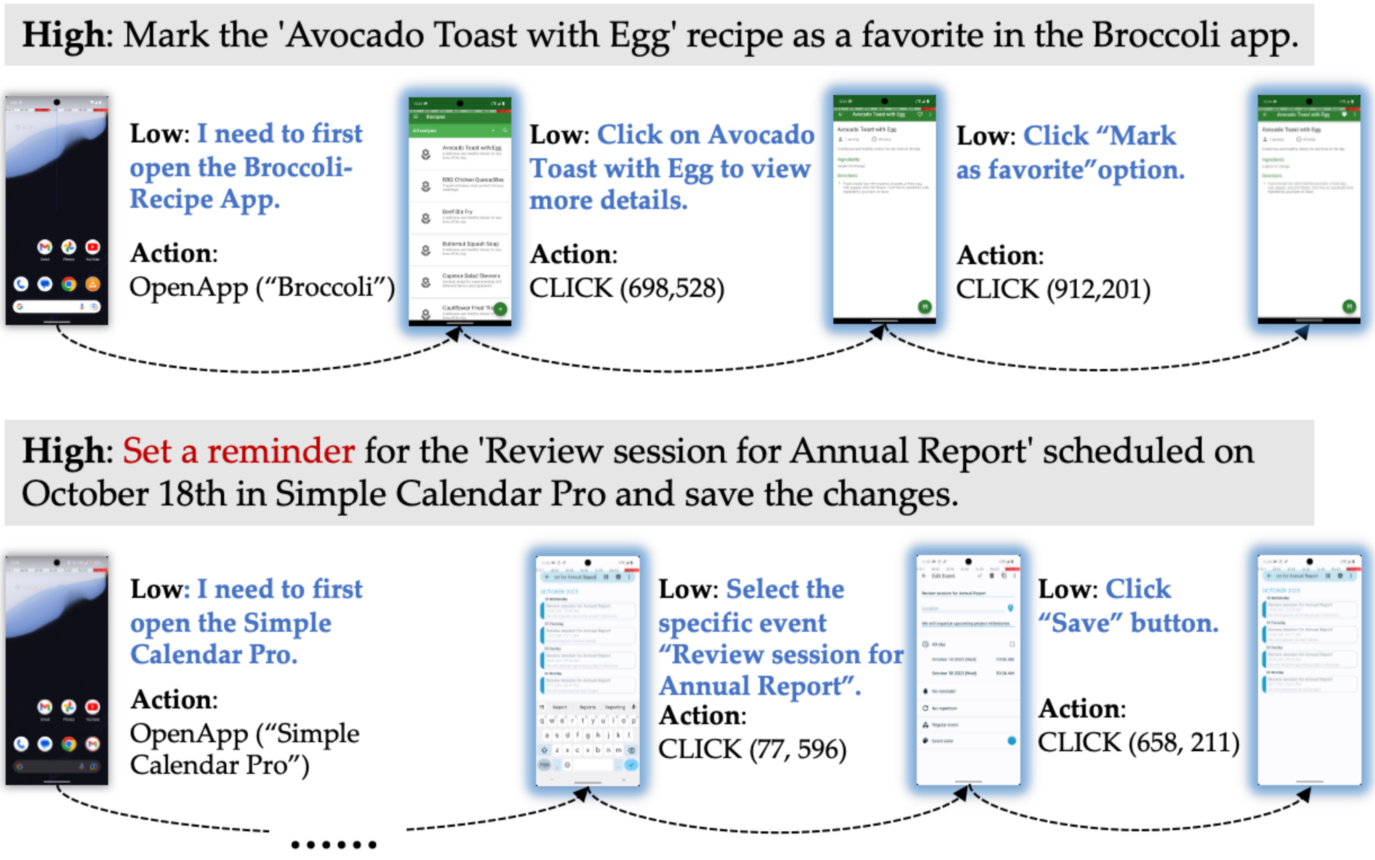
After reverse task synthesis generates task instructions, they are automatically executed in the GUI environment to build complete trajectories.

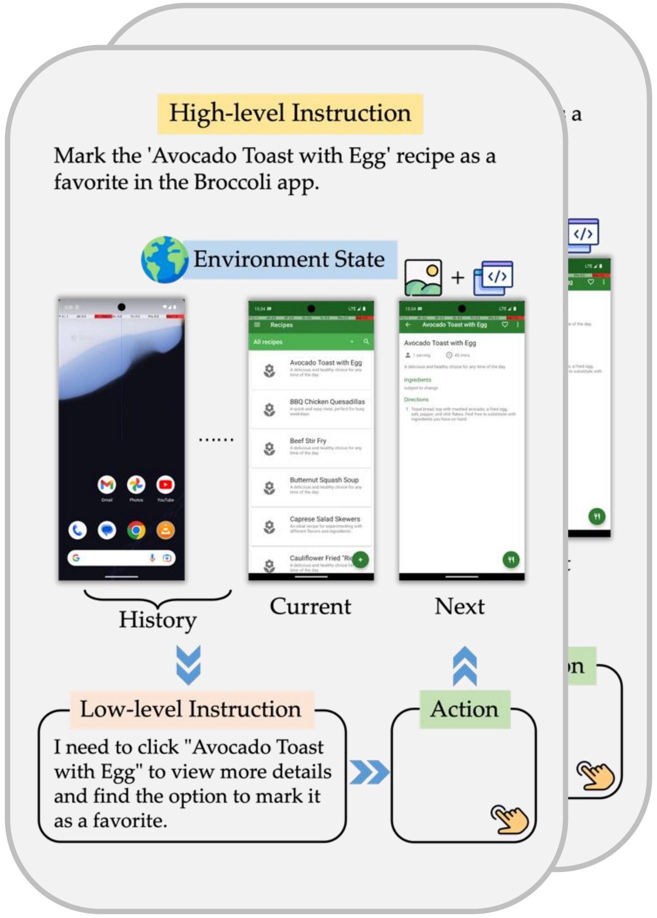


Reverse Task Synthesis

Trajectories collected! But is this all?

Let's consider data quality and synthesis efficiency.





Data Quality Control

Tasks are executed by machines, not all of them are successful.

Previous approach:

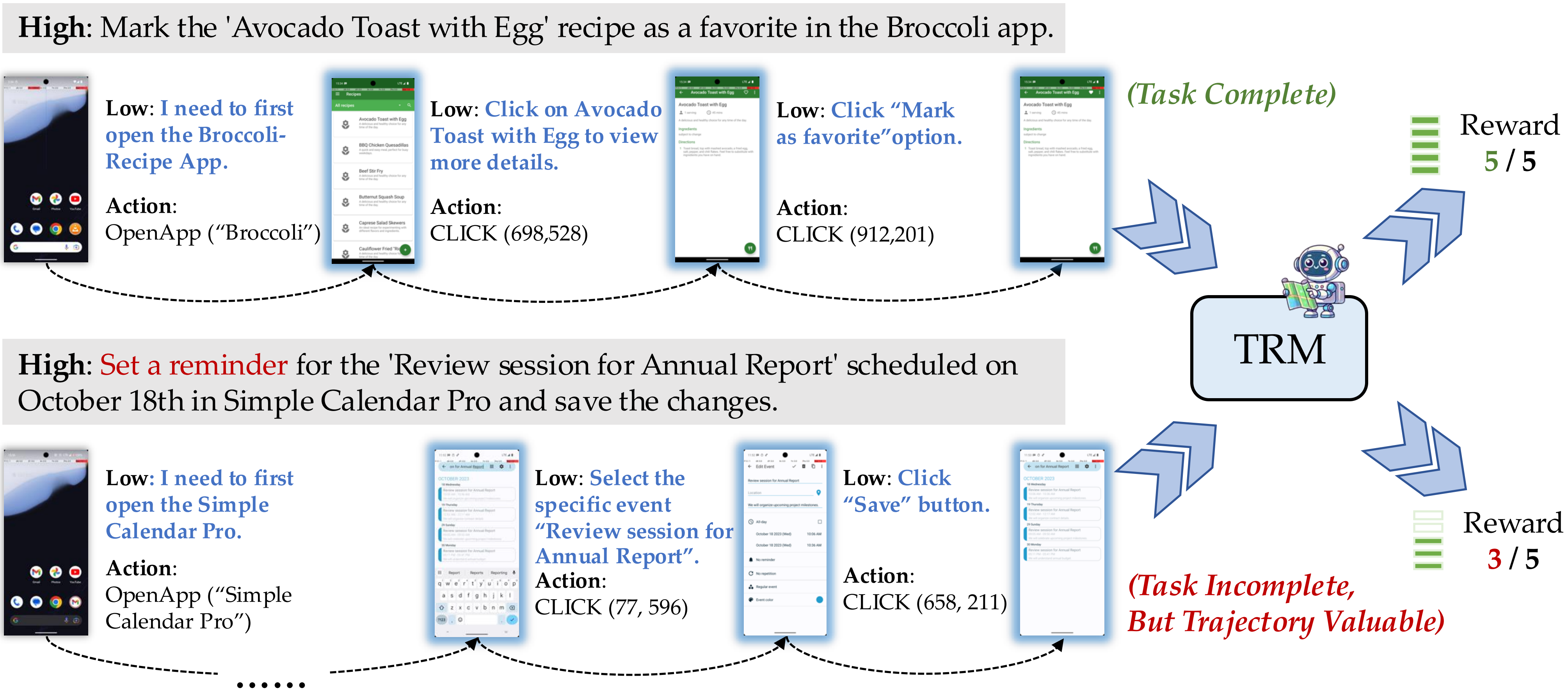
1. Training all data at once - what about the quality?

2. Discarding all incomplete Trajectories - what about the efficiency?

Thus, we introduce a Trajectory Reward Model to handle this.

Reward Modeling

We introduce a Trajectory Reward Model for weighted sampling in training.



Models

Data Synthesis

GPT-4o

Qwen2-VL-72B-Instruct

Backbones

InternVL2-4B / 8B

Qwen2-VL-7B-Instruct

Training Strategies

Leverage trajectory characteristics to train GUI agents with complete capabilities

1. Planning Training

High-level

Low-level

$$\mathcal{L}_1 = -\sum_{t_i \in \mathcal{T}} \log\Big(p_\theta(\ell \mid s, h_i, c)\cdot p_\theta(a \mid s, h_i, c, \ell)\Big)$$

Input

Actions

History

2. Action Training

$$\mathcal{L}_2 = -\sum_{t_i \in \mathcal{T}} \log p_\theta(a \mid s, c, \ell)$$

Training Strategies

After Training, our agents will generate ReACT-Style output

Examples:

Step 1: To create a new folder in Markor, I need to first open the Markor app.

action: `{"action_type":"open_app", "app_name":"Markor"}`

Step 2: To create a new folder, I need to click on the "Create a new file or folder" button, which is indicated by the plus icon.

action: `{"action_type":"click","x":964.5,"y":2074.5}`

Step 3: I need to change the folder name to folder_20241224. The current text field for the folder name is visible and editable.

action:
`{"action_type":"type","text":"folder_20241224","x":373.5,"y":552.0}`

…

Baselines

We adapt / build the following forward baselines

- **Zero-Shot.** Advanced prompting-based agents, such as M3A.

- **Task-Driven.** GUI Trajectories synthesized using pre-defined tasks. Given initial screenshots of the app/web page and task examples, use GPT-4 to generate high-level instructions and collect data.

- **Self-Instruct.** Builds on Task-Driven by adding self-instructed tasks.

Setting: Screenshot + A11ytree

[7] *Self-Instruct: Aligning Language Models with Self-Generated Instructions, ACL 2023*

Experiments: Mobile

Base Model	Strategies	AndroidWorld	AndroidControl-High		AndroidControl-Low	
			SR	Type	SR	Type
GPT-4o	Zero-Shot (M3A)	23.70	53.04	69.14	69.59	80.27
InternVL2-4B	Zero-Shot	0.00	16.62	39.96	33.69	60.65
	Task-Driven	4.02	27.37	47.08	66.48	90.37
	Task-Driven w. Self Instruct	7.14	24.95	44.27	66.70	90.79
	OS-Genesis	**15.18**	**33.39**	**56.20**	**73.38**	**91.32**
InternVL2-8B	Zero-Shot	2.23	17.89	38.22	47.69	66.67
	Task-Driven	4.46	23.79	43.94	64.43	89.83
	Task-Driven w. Self Instruct	5.36	23.43	44.43	64.69	89.85
	OS-Genesis	**16.96**	**35.77**	**64.57**	**71.37**	**91.27**
Qwen2-VL-7B	Zero-Shot	0.89	28.92	61.39	46.37	72.78
	Task-Driven	6.25	38.84	58.08	71.33	88.71
	Task-Driven w. Self Instruct	9.82	39.36	58.28	71.57	89.73
	OS-Genesis	**17.41**	**44.54**	**66.15**	**74.17**	**90.72**

Table 1: Performance on AndroidWorld and AndroidControl benchmarks.

Findings: OS-Genesis + Opensource VLM > Propriety Models + Complex Prompting

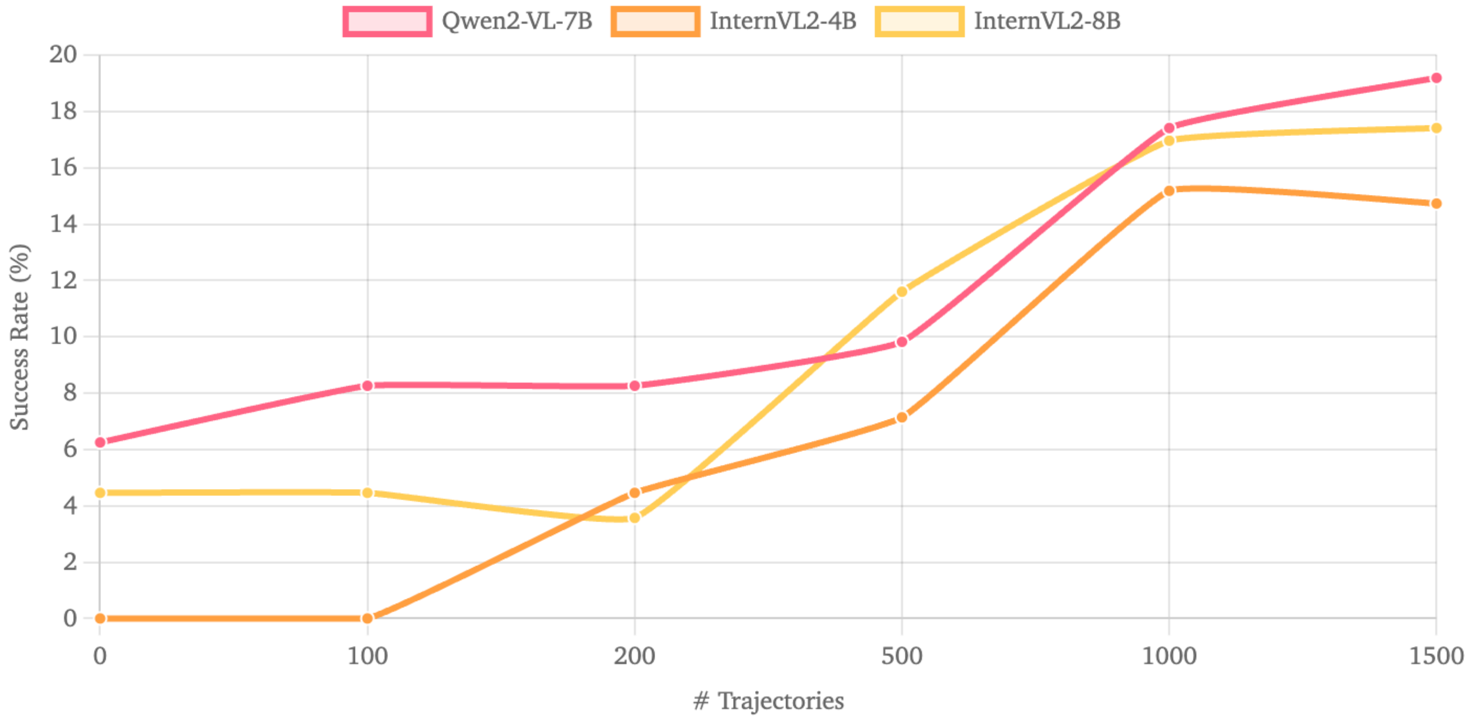
Experiments: Web

Base Model	Strategies	Shopping	CMS	Reddit	Gitlab	Maps	Overall
GPT-4o	Zero-Shot	14.28	21.05	6.25	14.29	20.00	16.25
InternVL2-4B	Zero-Shot	0.00	0.00	0.00	0.00	0.00	0.00
	Task-Driven	5.36	1.76	0.00	**9.52**	5.00	4.98
	Task-Driven w. Self Instruct	5.36	3.51	0.00	**9.52**	7.50	5.81
	OS-Genesis	**10.71**	**7.02**	**3.13**	7.94	**7.50**	**7.88**
InternVL2-8B	Zero-Shot	0.00	0.00	0.00	0.00	0.00	0.00
	Task-Driven	3.57	7.02	0.00	6.35	2.50	4.56
	Task-Driven w. Self Instruct	**8.93**	10.53	6.25	**7.94**	0.00	7.05
	OS-Genesis	7.14	**15.79**	**9.34**	6.35	**10.00**	**9.96**
Qwen2-VL-7B	Zero-Shot	**12.50**	7.02	6.25	6.35	5.00	7.47
	Task-Driven	8.93	7.02	6.25	6.35	5.00	7.05
	Task-Driven w. Self Instruct	8.93	1.76	3.13	4.84	**7.50**	5.39
	OS-Genesis	7.14	**8.77**	**15.63**	**15.87**	5.00	**10.79**

Table 2: Performance on WebArena benchmarks.

Analysis

How Scaling Trajectory Data Improves Agentic Ability?

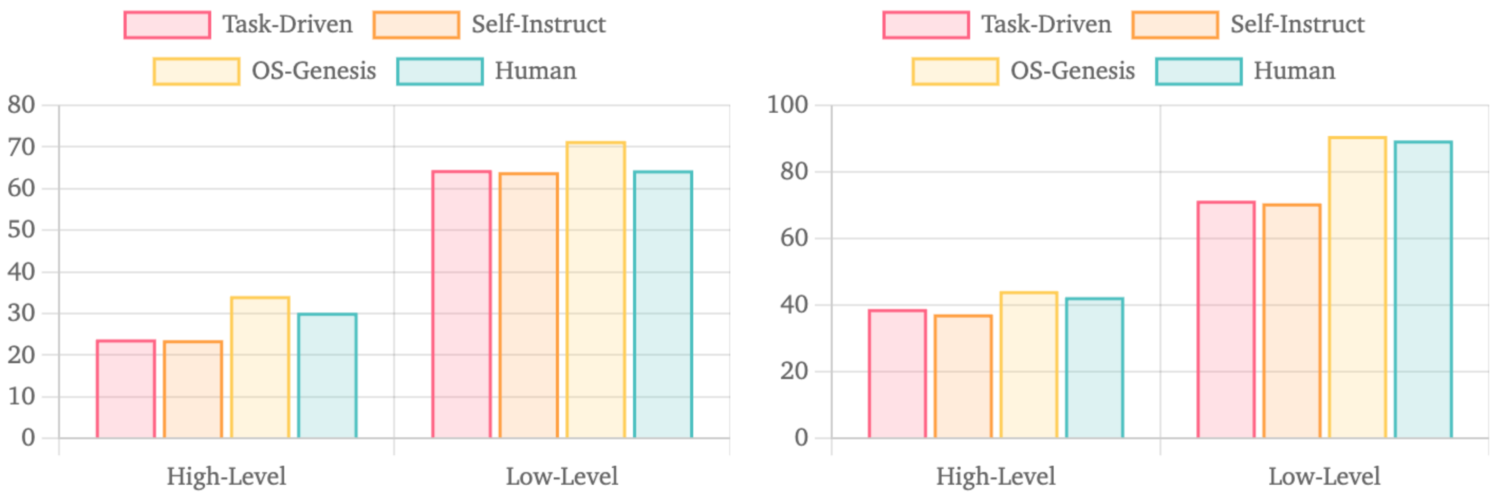


Insight: Generally improves, but will saturate.

Analysis

How Far are we from Human Data?

Let's first take a look at high-level instructions.



Insight: Reverse Task Synthesis Elicits Better Executability.

Analysis

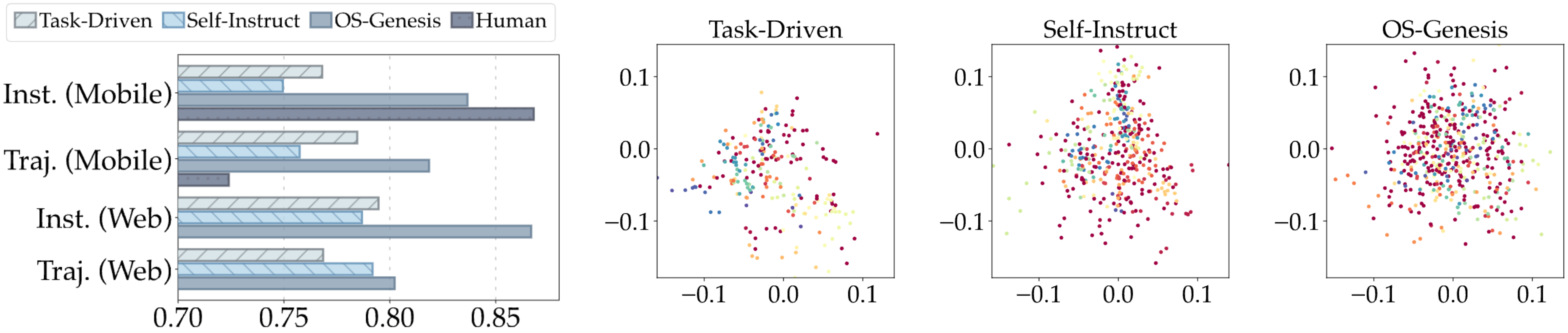
How Far are we from Human Data?

Then, OS-Genesis v.s. Human-annotated Trajectories.

Insight: OS-Genesis achieves ~80% of human data's effectiveness.

Analysis

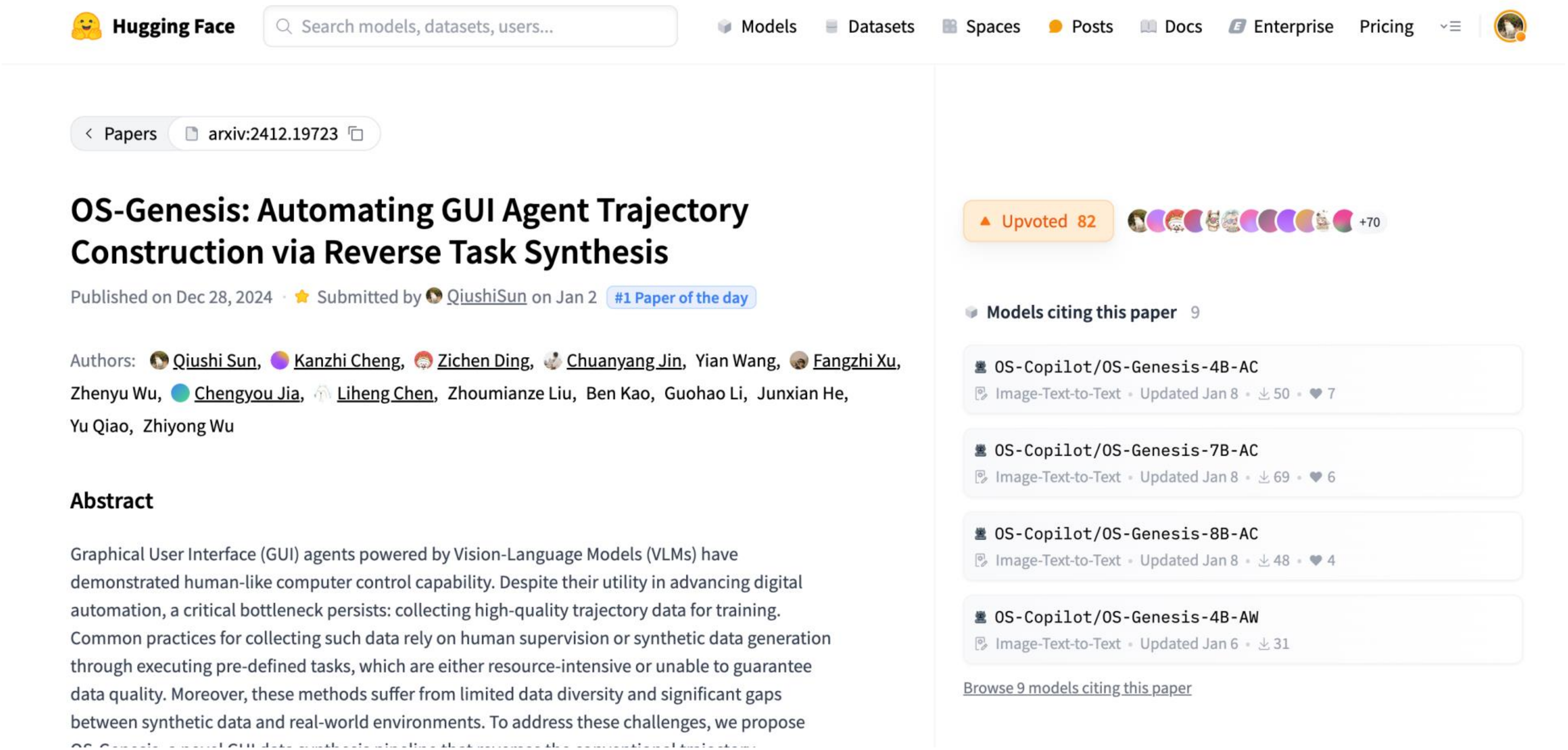
How about our data diversity?



Insight: Significantly better than Forward methods and approaches the human level.

Checkpoints & Data Access

Available on Hugging Face



Checkpoints & Data Access

Available on ModelScope

Our Project

OS-Genesis

Automating GUI Agent Trajectory Construction via Reverse Task Synthesis

Introducing OS-Genesis, a ***manual-free*** data pipeline for synthesizing GUI agent trajectory. OS-Genesis is characterized by the following core features:

- **Interaction-driven**: Agents actively explore GUI environments through stepwise interactions to discover functionalities and generate data.
- **Reverse Task Synthesis**: OS-Genesis retroactively derives meaningful low/high-level task instructions from observed interactions and state changes, enabling the construction of diverse and executable trajectories without pre-defined tasks.
- **Trajectory Data**: We construct and release high-quality mobile and web trajectories to accelerate GUI agents research.
- **Performance**: OS-Genesis significantly outperforms other synthesis methods on benchmarks like AndroidWorld and WebArena.

arXiv | Code | Checkpoints | Data

https://qiushisun.github.io/OS-Genesis-Home/

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

SCHOOL OF COMPUTING & DATA SCIENCE
The University of Hong Kong

Thanks for listening

Contact: qiushisun@connect.hku.hk