



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



SCHOOL OF
COMPUTING &
DATA SCIENCE
The University of Hong Kong



NICE

NLP Academic
Exchange Platform

ACL 2025
VIENNA

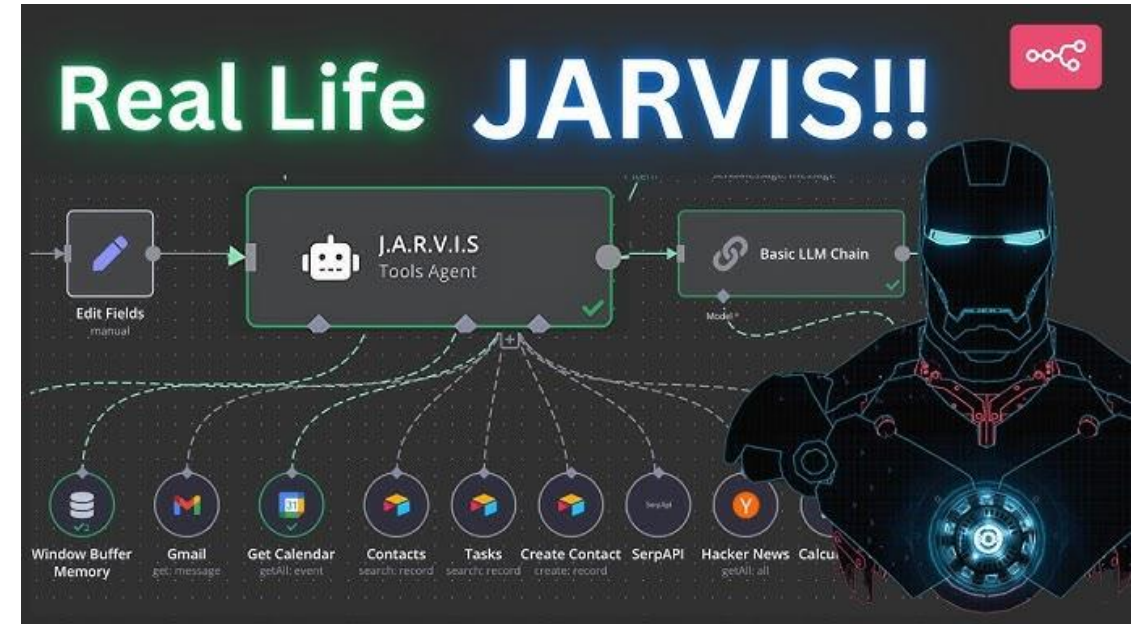
OS-Genesis: Automating GUI Agent Trajectory Construction via Reverse Task Synthesis

Qiushi Sun

qiushisun.github.io

✉ [@qiushi_sun](https://twitter.com/qiushi_sun)

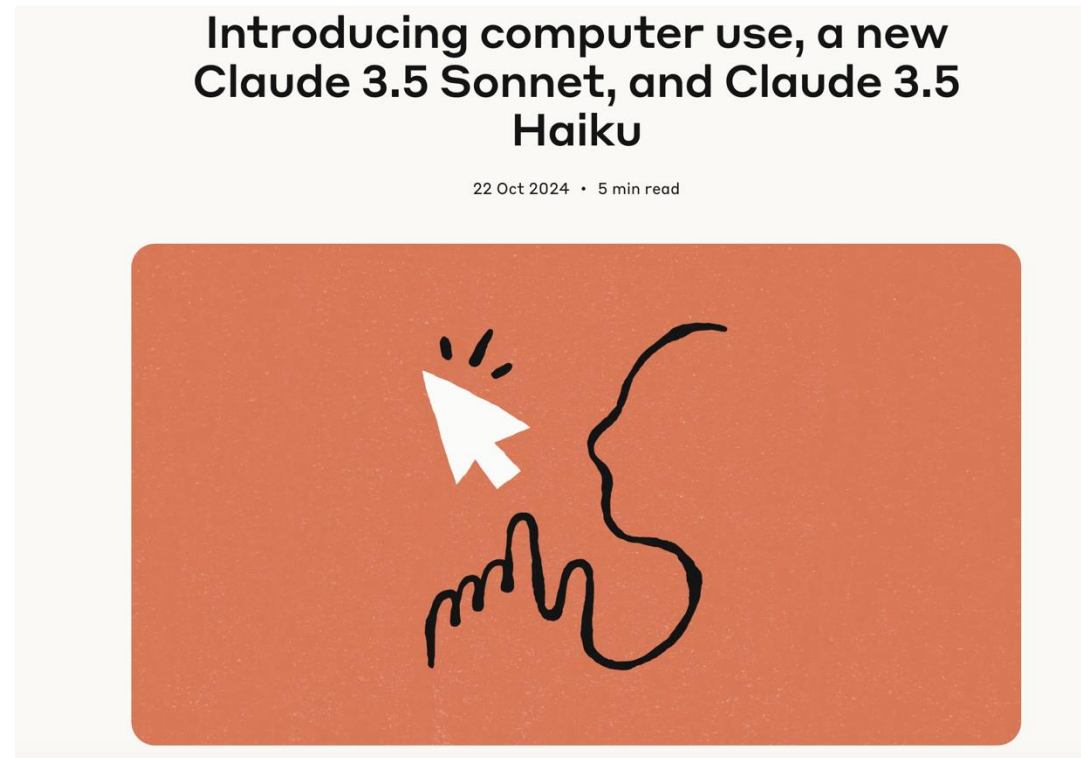
Computer Use Agents



The Feasibility of Jarvis AI from Marvel in Real Life

Computer Use Agents

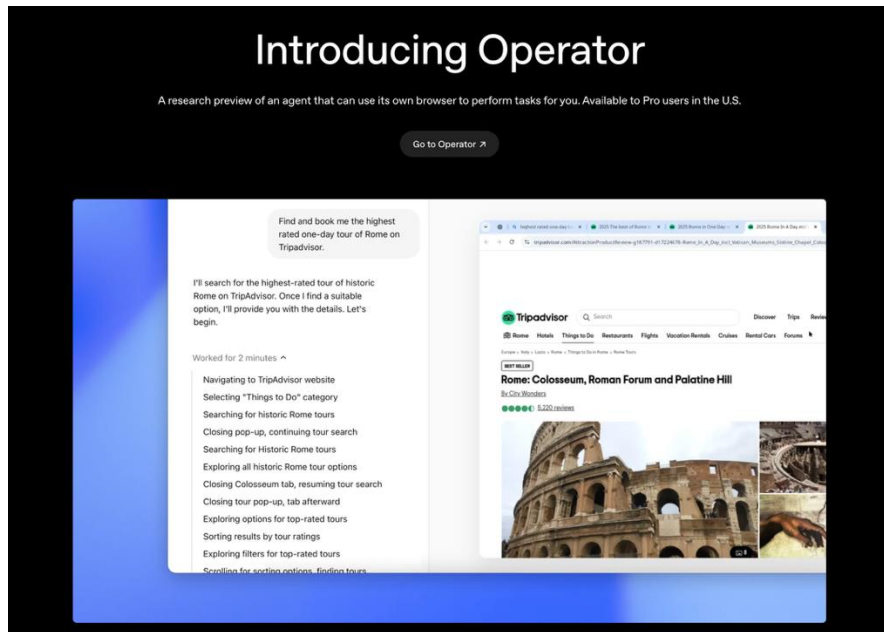
Both academia and industry are building **computer use agents**



Claude Computer Use

Computer Use Agents

Automating daily computer tasks

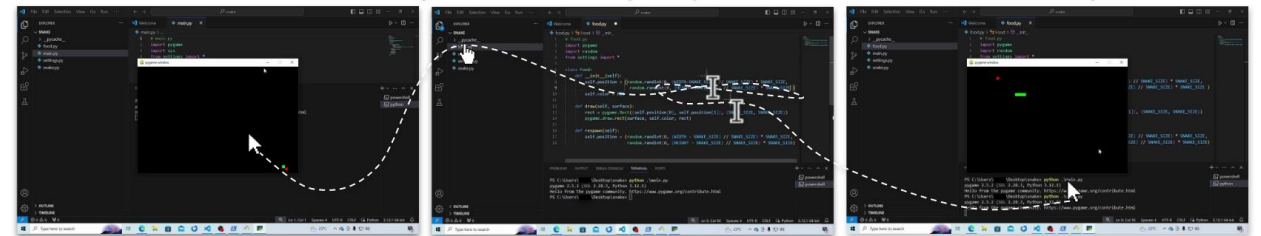


OpenAI Operator

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



Task instruction 2: ...some details about snake game omitted... Could you help me tweak the code so the snake can actually eat the food?



Daily Computer Use

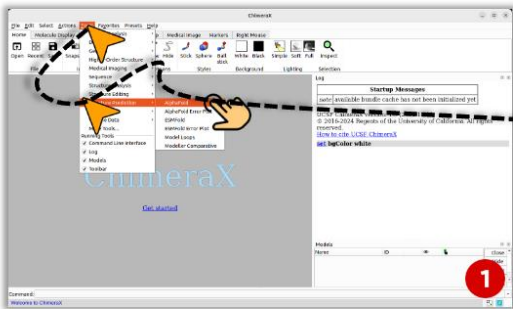
[3] *Introducing Operator: A research preview of an agent that can use its own browser to perform tasks for you.*, Jan 23, 2025

[4] *OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments*

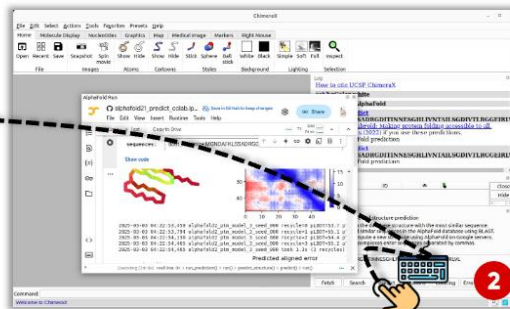
Computer Use Agents

Automate scientific workflows

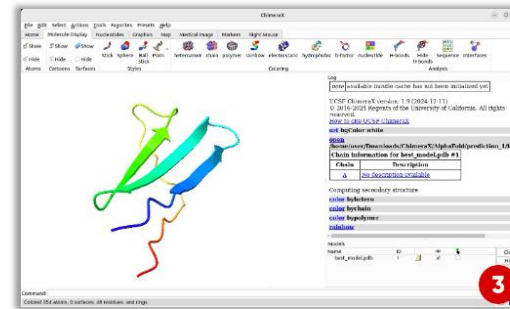
Instruction: Predict the protein structure for the amino acid sequence of 'MGND...' via AlphaFold in ChimeraX.



Step1: Toggle the widget of AlphaFold.

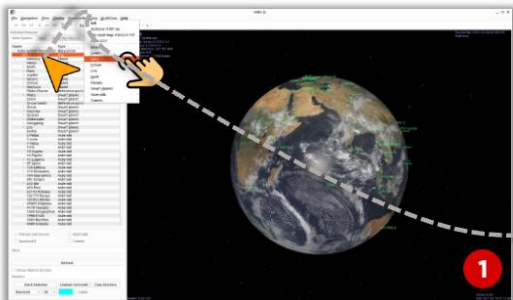


Step2: Input the given sequence and call out AlphaFold for structure prediction.

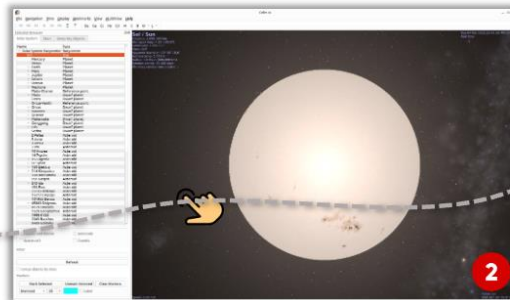


Step3: Wait until the prediction finished.

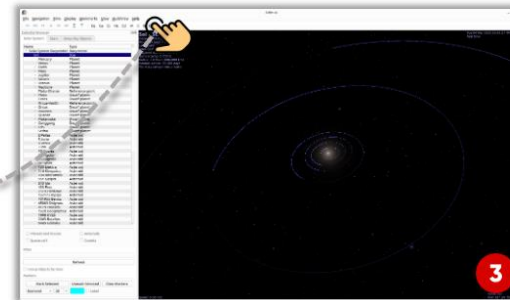
Instruction: Show planets' orbits of Solar System in Celestia.



Step1: Select the Sol and click 'Goto' in context menu.



Step2: Slide the mouse wheel to move the camera away from Sol.



Step3: Click to show orbits of planets.

Some Typical & Recent works



SeeClick: Harnessing GUI Grounding for Advanced Visual GUI Agents, [ACL 2024](#)



OS-ATLAS: A Foundation Action Model for Generalist GUI Agents , [ICLR 2025 Spotlight](#)



OS-Genesis: Automating GUI Agent Trajectory Construction via Reverse Task Synthesis , [ACL 2025](#)



Breaking the Data Barrier -- Building GUI Agents Through Task Generalization



AgentStore: Scalable Integration of Heterogeneous Agents As Specialized Generalist Computer Assistant , [ACL 2025](#)



ScienceBoard: Evaluating Multimodal Autonomous Agents in Realistic Scientific Workflows

Some Typical & Recent works

SeeClick: Harnessing GUI Grounding for Advanced Visual GUI Agents, ACL 2024

OS-ATLAS: A Foundation Action Model for Generalist GUI Agents , ICLR 2025 Spotlight



OS-Genesis: Automating GUI Agent Trajectory Construction via Reverse Task Synthesis , **ACL 2025**



Breaking the Data Barrier -- Building GUI Agents Through Task Generalization

AgentStore: Scalable Integration of Heterogeneous Agents As Specialized Generalist Computer Assistant , ACL 2025

ScienceBoard: Evaluating Multimodal Autonomous Agents in Realistic Scientific Workflows

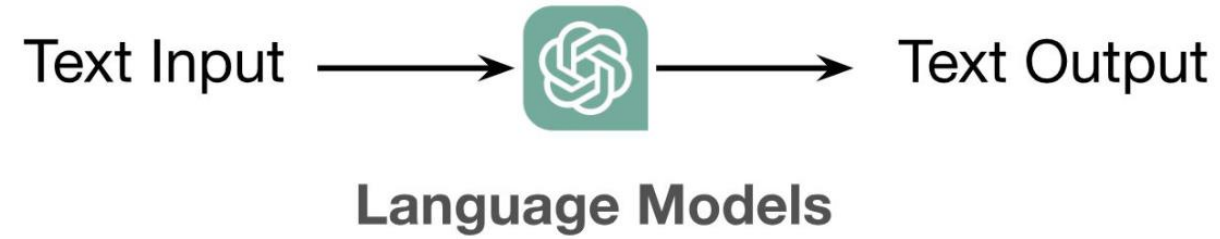
Build Computer Use Agents

They are quite promising for achieving Digital Automation through CLI or GUI.

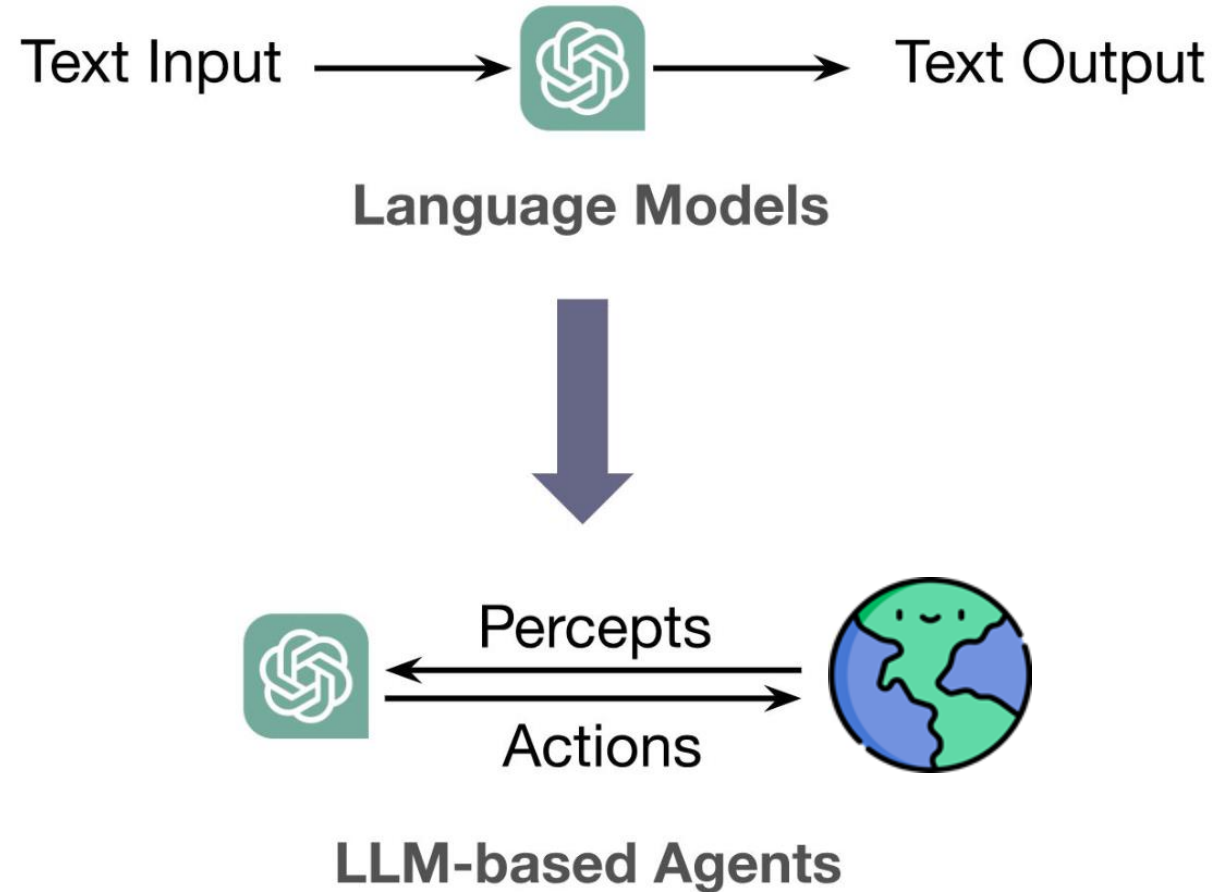
Can we transform a (V)LM into such GUI agents?

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 Problems

Human annotation for GUI data is **much more expensive** than you think. 

Not to mention scenario/domain - specific data.

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 for GUI agents

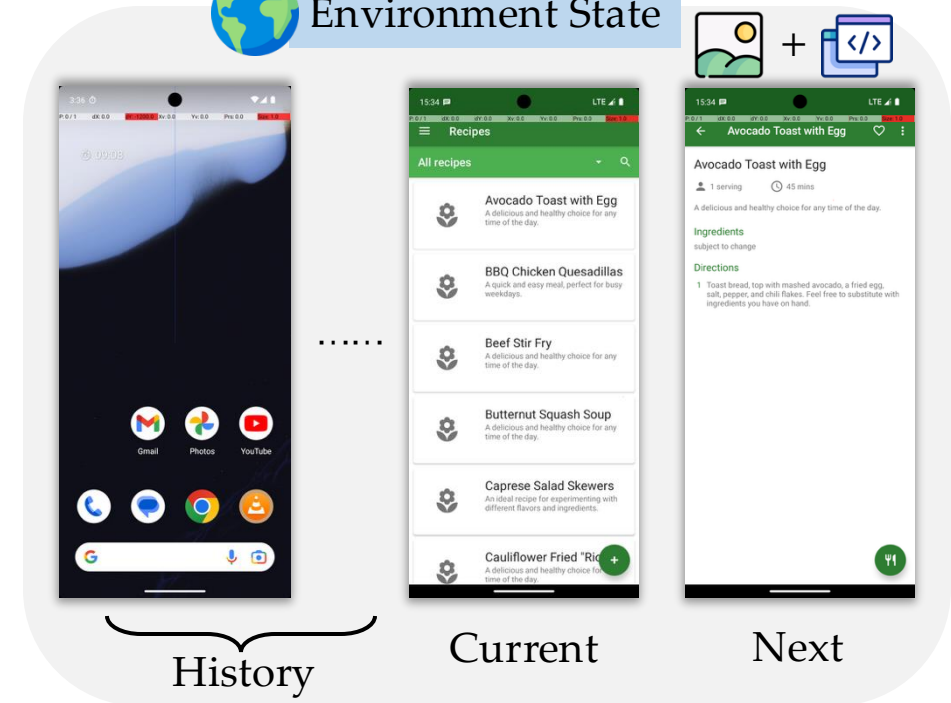
1. A **high-level instruction** that defines the overall goal the agent aims to accomplish
2. A series of **low-level instructions** that each describe specific steps required
3. **Actions** (e.g., CLICK, TYPE) 
4. **States**, which include visual representations like screenshots and textual representations such as a11ytree 

High-level Instruction

Mark the 'Avocado Toast with Egg' recipe as a favorite in the Broccoli app.



Environment State



Low-level Instruction

I need to click "Avocado Toast with Egg" to view more details and find the option to mark it as a favorite.

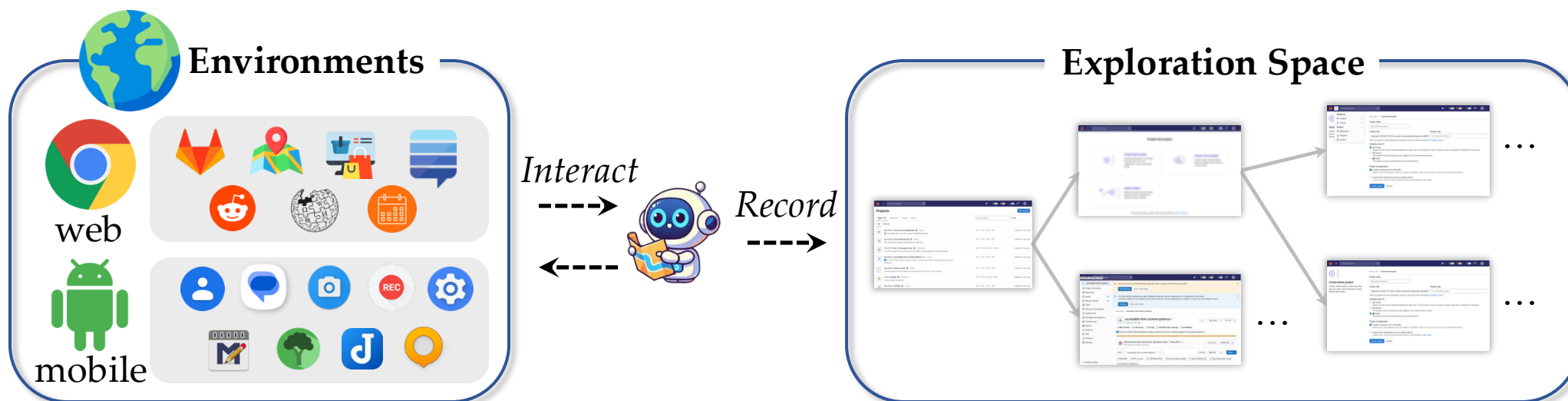
Action

CLICK [Avocado Toast with Egg]
(698, 528)

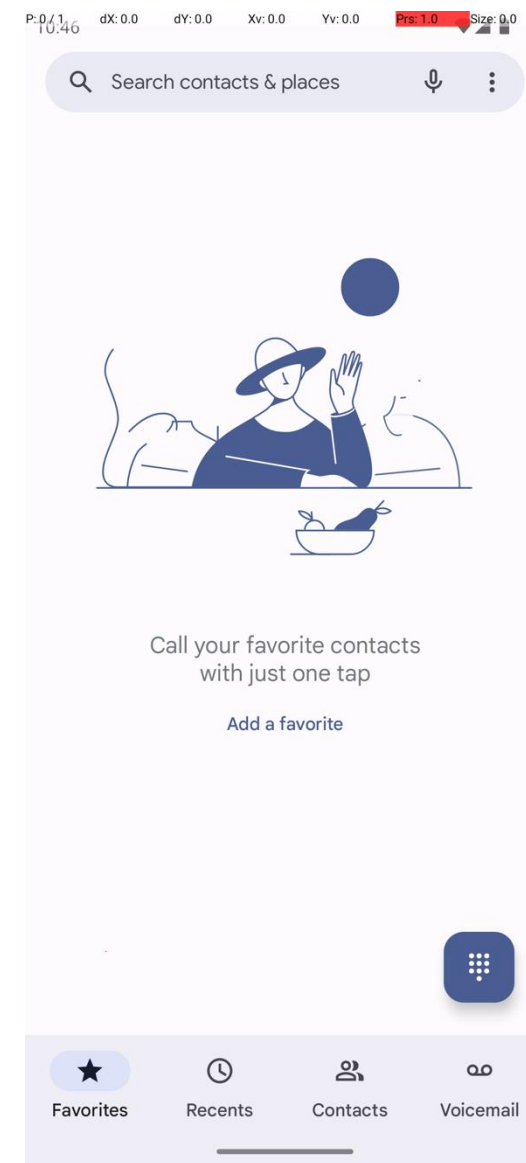
Reverse Task Synthesis

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: $\langle \text{Screen1}, \text{action}, \text{Screen2} \rangle$



Dynamic Environments





Dynamic Environments



My Account My Wish List Sign Out Welcome to One Stop Market

One Stop Market

Search entire store here...  

[Advanced Search](#)



Beauty & Personal Care - Sports & Outdoors - Clothing, Shoes & Jewelry - Home & Kitchen - Office Products - Tools & Home Improvement -


Health & Household - Patio, Lawn & Garden - Electronics - **Cell Phones & Accessories** - Video Games - Grocery & Gourmet Food -

Home > Cell Phones & Accessories

Cell Phones & Accessories

Shop By

  Items 1-12 of 2449

Sort By Position 

Shopping Options

Category

[Accessories\(1924\)](#)

[Cases, Holsters & Sleeves\(457\)](#)


[Cell Phones\(68\)](#)

Price

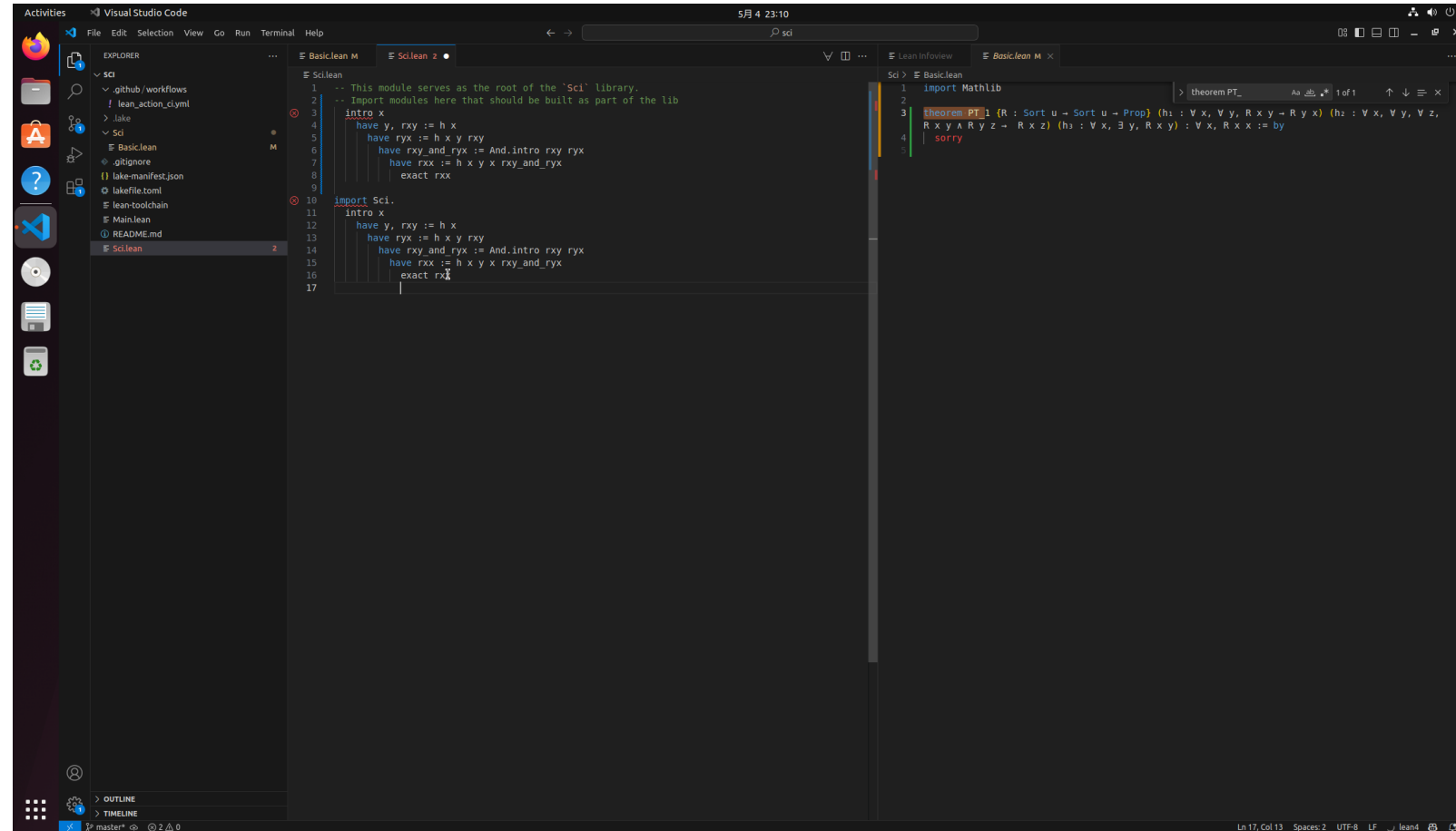
[\\$0.00 - \\$999.99\(2446\)](#)

[\\$1,000.00 and above\(3\)](#)

[Compare Products](#)



Dynamic Environments

A screenshot of the Visual Studio Code editor interface. The left sidebar shows the Explorer view with a file tree containing folders like 'github/workflows', 'lake', 'Sci', and files like 'lean_action_ci.yml', 'lake-manifest.json', 'lakefile.toml', 'lean-toolchain', 'Main.lean', and 'README.md'. The main editor area is split into two panes. The left pane shows a Lean 4 file named 'Sci.lean' with the following code:

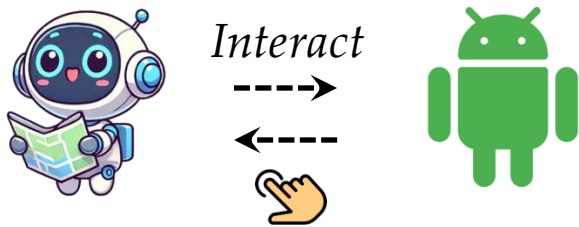
```
1 -- This module serves as the root of the 'Sci' library.
2 -- Import modules here that should be built as part of the lib
3
4 intro x
5   have y, rxy := h x
6   have rxy_and_rxy := And.intro rxy rxy
7   have rxx := h x y x rxy_and_rxy
8   exact rxx
9
10 import Sci.
11 intro x
12   have y, rxy := h x
13   have rxy_and_rxy := And.intro rxy rxy
14   have rxx := h x y x rxy_and_rxy
15   exact rxx
16
17
```

The right pane shows a Lean 4 file named 'Basic.lean' with the following code:

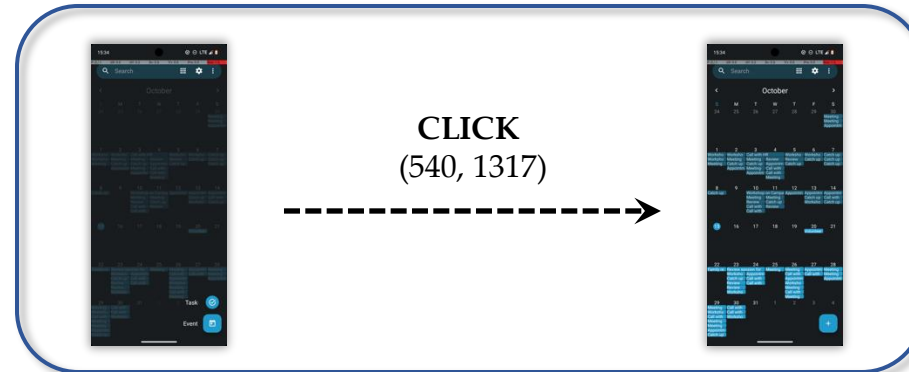
```
1 import Mathlib
2
3 theorem PT_1 (R : Sort u → Sort u → Prop) (h1 : ∀ x, ∀ y, R x y → R y x) (h2 : ∀ x, ∀ y, ∀ z,
4   R x y ∧ R y z → R x z) (h3 : ∀ x, ∃ y, R x y) : ∀ x, R x x := by
5   sorry
```


Reverse Task Synthesis

Retroactively interpreting changes in the GUI environment caused by actions.

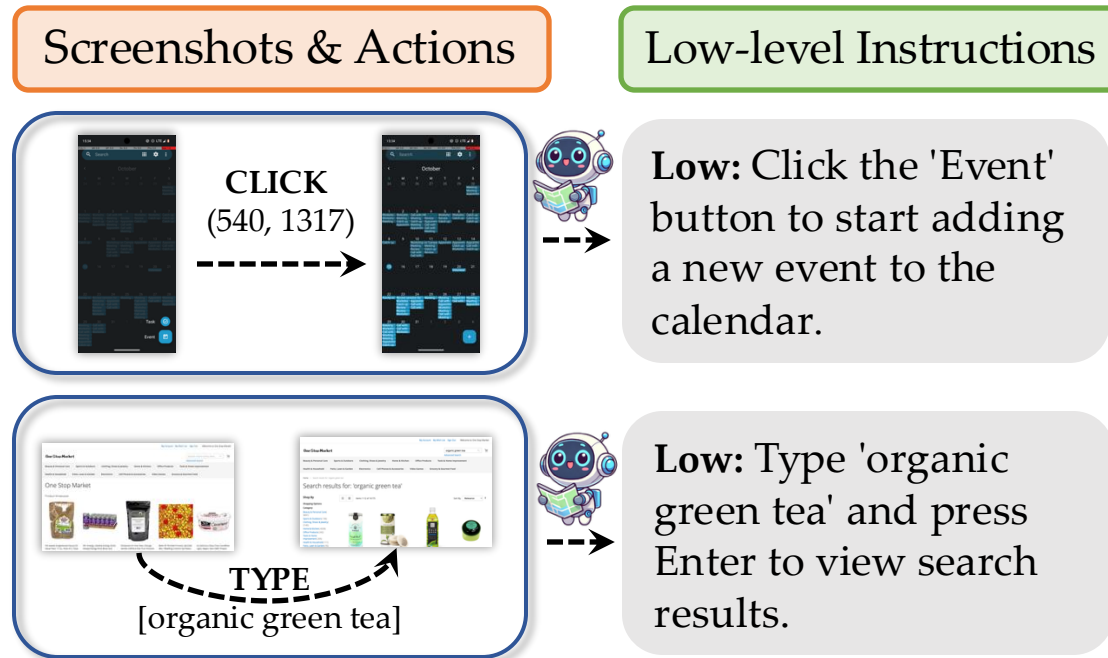


Screenshots & Actions



Reverse Task Synthesis

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

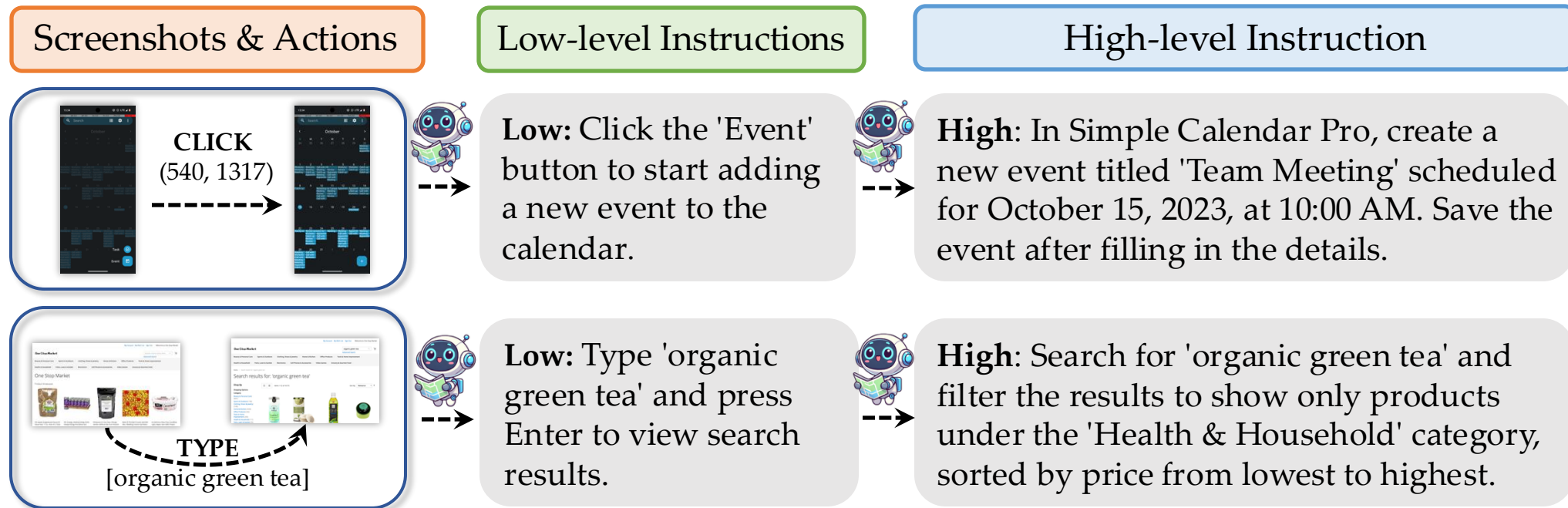


The data we synthesized:

1. Grounded
2. Actionable

Reverse Task Synthesis

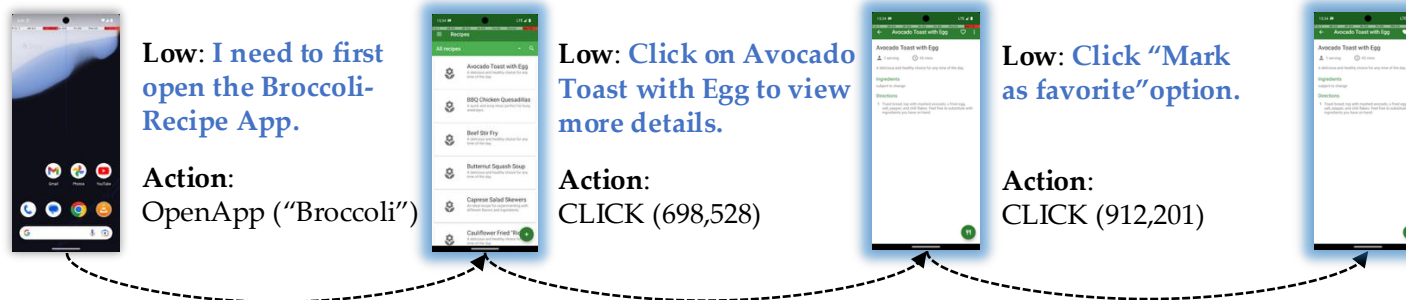
Retroactively interpreting changes in the GUI environment caused by actions, this process generates executable low-level instructions, which are then transformed into broader, goal-oriented high-level tasks



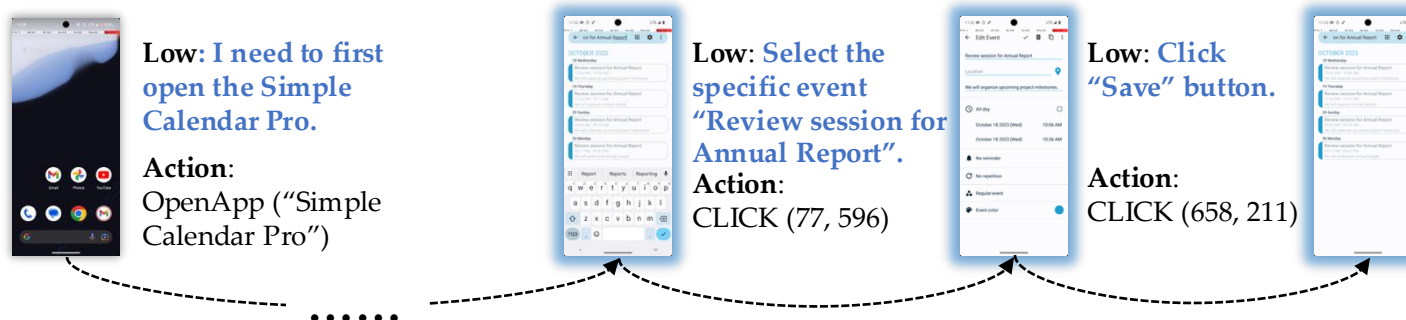
Reverse Task Synthesis

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

High: Mark the 'Avocado Toast with Egg' recipe as a favorite in the Broccoli app.



High: Set a reminder for the 'Review session for Annual Report' scheduled on October 18th in Simple Calendar Pro and save the changes.

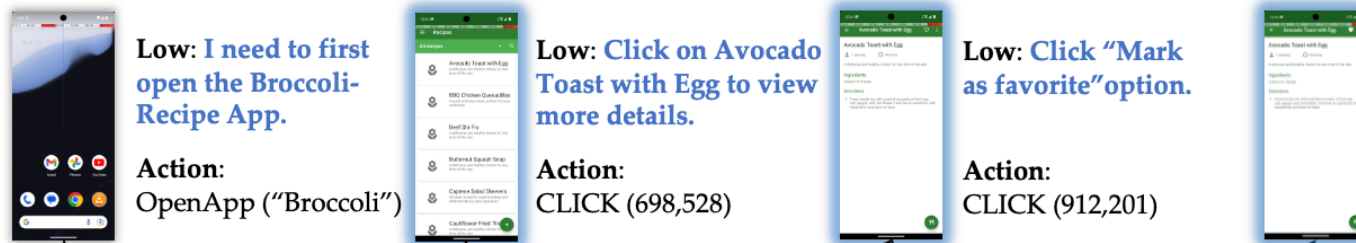


Reverse Task Synthesis

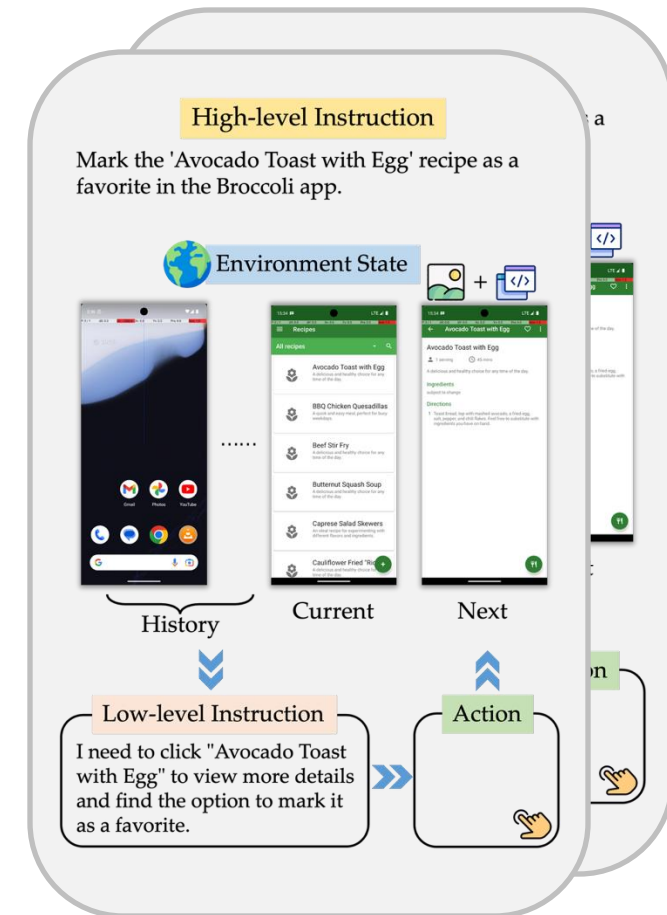
Trajectories collected! But is this all?

Let's consider data **quality** and synthesis **efficiency**.

High: Mark the 'Avocado Toast with Egg' recipe as a favorite in the Broccoli app.



High: Set a reminder for the 'Review session for Annual Report' scheduled on October 18th in Simple Calendar Pro and save the changes.



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.

High: Mark the 'Avocado Toast with Egg' recipe as a favorite in the Broccoli app.



High: Set a reminder for the 'Review session for Annual Report' scheduled on October 18th in Simple Calendar Pro and save the changes.



Models

Data Synthesis



GPT-4o



Qwen-VL

Qwen2-VL-72B-Instruct

Backbones



InternVL

InternVL2-4B / 8B



Qwen-VL

Qwen2-VL-7B-Instruct

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

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 Far are we from **Human Data**?

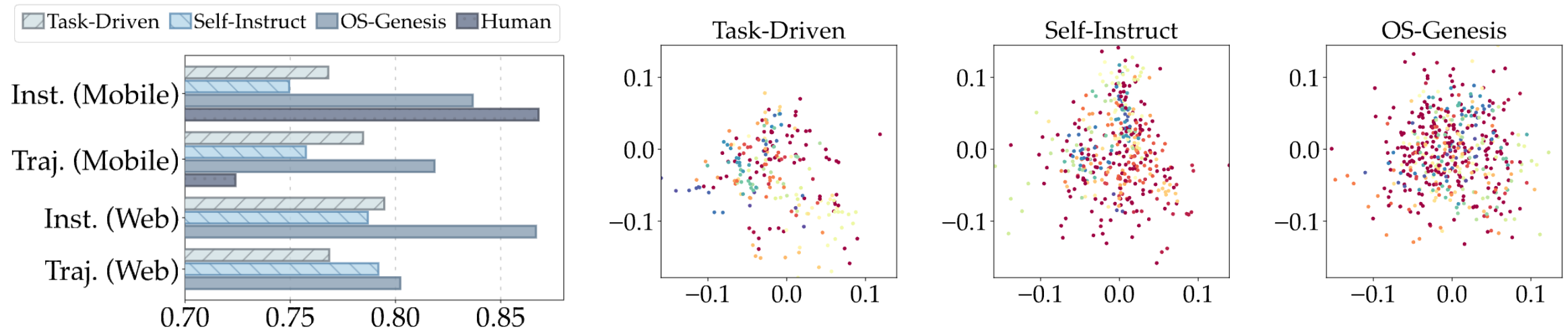
Then, OS-Genesis v.s. **Human-annotated Trajectories**.



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

 **Hugging Face**

Models

Datasets


Spaces

Posts

Docs

Enterprise


Pricing










< Papers

arxiv:2412.19723

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


Published on Dec 28, 2024 · ★ Submitted by  [Qiushi Sun](#) on Jan 2 #1 Paper of the day

Authors:  [Qiushi Sun](#),  [Kanzhi Cheng](#),  [Zichen Ding](#),  [Chuanyang Jin](#), Yian Wang,  [Fangzhi Xu](#), Zhenyu Wu,  [Chengyou Jia](#),  [Liheng Chen](#), Zhoudmianze Liu, Ben Kao, Guohao Li, Junxian He, Yu Qiao, Zhiyong Wu


Abstract


Graphical User Interface (GUI) agents powered by Vision-Language Models (VLMs) have demonstrated human-like computer control capability. Despite their utility in advancing digital automation, a critical bottleneck persists: collecting high-quality trajectory data for training. Common practices for collecting such data rely on human supervision or synthetic data generation through executing pre-defined tasks, which are either resource-intensive or unable to guarantee data quality. Moreover, these methods suffer from limited data diversity and significant gaps between synthetic data and real-world environments. To address these challenges, we propose OS-Genesis, a novel GUI data synthesis framework that reverses the conventional trajectory


▲ Upvoted 82


 +70

Models citing this paper 9

 OS-Copilot/OS-Genesis-4B-AC
Image-Text-to-Text · Updated Jan 8 · 📄 50 · ❤️ 7

 OS-Copilot/OS-Genesis-7B-AC
Image-Text-to-Text · Updated Jan 8 · 📄 69 · ❤️ 6

 OS-Copilot/OS-Genesis-8B-AC
Image-Text-to-Text · Updated Jan 8 · 📄 48 · ❤️ 4

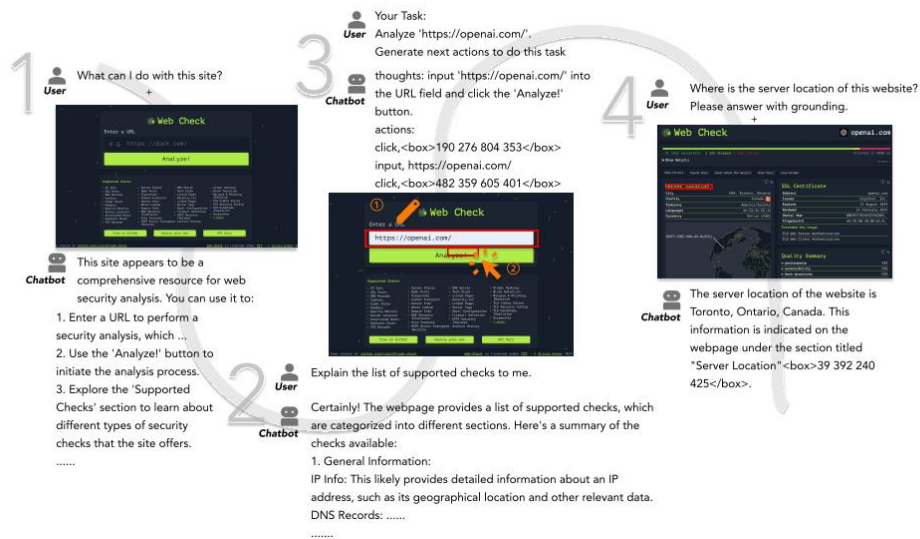
 OS-Copilot/OS-Genesis-4B-AW
Image-Text-to-Text · Updated Jan 6 · 📄 31

[Browse 9 models citing this paper](#)

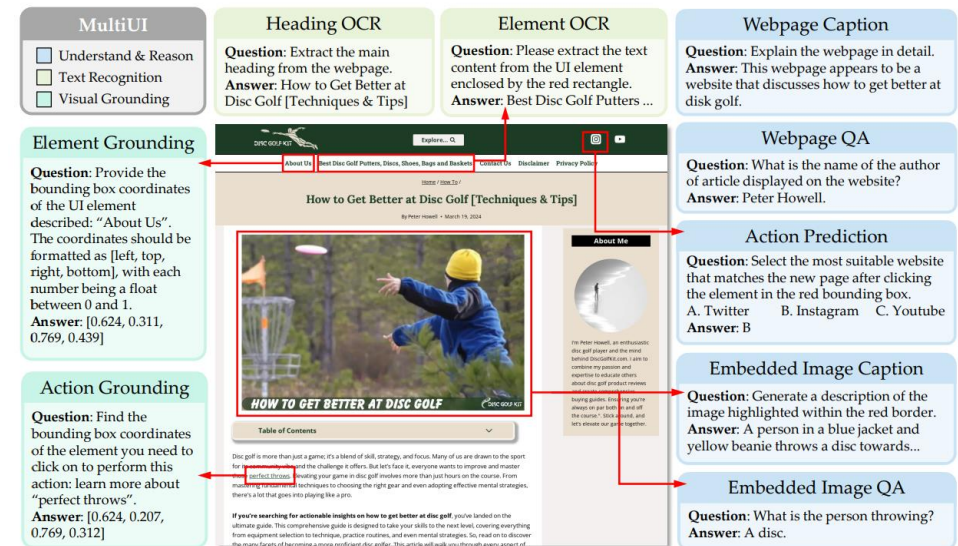
Page 34

Next Step?

Beyond Planning and Action: What Else Do We Need?



GUI Trajectory Data



GUI Perception/Knowledge...

[9] *Harnessing Webpage UIs for Text-Rich Visual Understanding*, ICLR 2025.

[10] *Guicourse: From general vision language models to versatile gui agents*, arxiv 2406.11317

GUIMid

Beyond Planning and Action: What Else Do We Need?

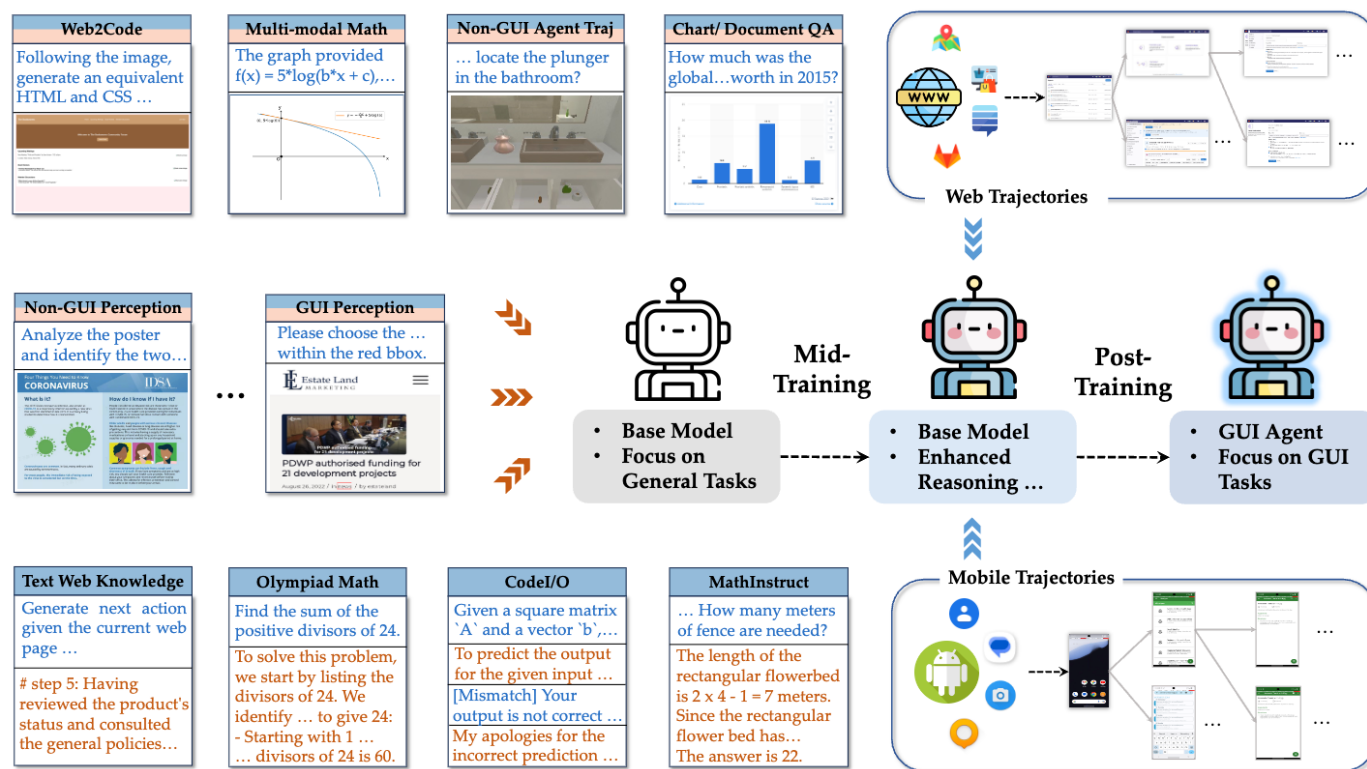
Domains	Ability	Datasets	Samples	Type
Vision-and-Language Modality				
Chart/Document QA	Perception	InfographicVQA (Guo et al., 2024)	2,184	Instruction, Thought*, Answer
		Ureader QA (Guo et al., 2024)	53,794	Instruction, Thought, Answer
		MPDocVQA (Tito et al., 2023)	431	Instruction, Thought, Answer
		MathV360k (Liu et al., 2024b)	93,591	Instruction, Thought, Answer
Non-GUI Perception	Perception	Ureader OCR (Ye et al., 2023)	6,146	Instruction, Thought*, Answer
		DUE (Borchmann et al., 2021)	143,854	Instruction, Answer
GUI Perception	Perception	MultiUI (Liu et al., 2024a)	150,000	Instruction, Answer
Web Screenshot2Code	Perception	Web2Code (Yun et al., 2024)	150,000	Instruction, Answer
Multi-modal Math	Reasoning	Mavis (Zhang et al., 2024b)	150,000	Instruction, Thought, Answer
Multi-round Visual Conversation	Interaction	SVIT (Zhao et al., 2023)	150,000	Instruction, Thought, Answer
Non-GUI Agent Trajectories	Interaction	AlfWorld (Guo et al., 2024)	51,780	Instruction, Thought, Answer
Language Modality				
MathInstruct	Reasoning	MathInstruct (Yue et al., 2023)	150,000	Instruction, Thought, Answer
Olympiad Math	Reasoning	NuminaMath (LI et al., 2024)	150,000	Instruction, Thought, Answer
CodeI/O	Reasoning	CodeI/O (Li et al., 2025)	150,000	Instruction, Thought, Answer
Web Knowledge Base	Knowledge	Synatra (Ou et al., 2024)	99,924	Instruction, Thought, Answer
		AgentTrek (Xu et al., 2024a)	50,076	Instruction, Thought, Answer

We have abundant non-GUI data available to enhance versatile abilities

Can we take advantage of these data-rich domains?

GUIMid

Breaking the Data Barrier – Building GUI Agents Through Task Generalization: arXiv 2504.10127



Mid-training: Training phrase **between** pre-training and post-training

GUIMid

Breaking the Data Barrier – Building GUI Agents Through Task Generalization:
arXiv 2504.10127

Domains	Observation	WebArena		AndroidWorld
		PR	SR	SR
GUI Post-Training Only	Image	26.3	6.2	9.0
Public Baselines				
GPT-4o-2024-11-20	Image	36.9	15.6	11.7
OS-Genesis-7B	Image + Accessibility Tree	-	-	17.4
AGUVIS-72B	Image	-	-	26.1
Claude3-Haiku	Accessibility Tree	26.8	12.7	-
Llama3-70b	Accessibility Tree	35.6	12.6	-
Gemini1.5-Flash	Accessibility Tree	32.4	11.1	-
Vision-and-Language Modality				
Chart/ Document QA	Image	24.6	6.2	15.3
Non-GUI Perception	Image	28.7	7.6	14.0
GUI Perception	Image	27.4	7.1	14.0
Web Screenshot2Code	Image	28.0	6.6	9.9
Non-GUI Agents	Image	30.8	8.5	13.5
Multi-modal Math ✓	Image	30.4	8.5	15.3
Multi-round Visual Conversation	Image	30.0	9.0	12.6
Language Modality				
MathInstruct ✓	Image	31.9	10.9	14.4
Olympiad Math ✓	Image	31.5	8.5	13.1
CodeI/O ✓	Image	29.2	9.0	14.9
Web Knowledge Base	Image	31.3	9.5	9.0
Domain Combination (Sampled data from ✓ domains)				
GUIMid	Image	34.3	9.5	21.2





Mid-training: Training phrase **between** pre-training and post-training

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





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



SCHOOL OF
COMPUTING &
DATA SCIENCE
The University of Hong Kong



NICE

NLP Academic
Exchange Platform

ACL 2025
VIENNA

Thanks for listening

Contact: qiushisun@connect.hku.hk