

# **Part III:** Applications, Data, and Evaluation

Tao Yu

# Agenda

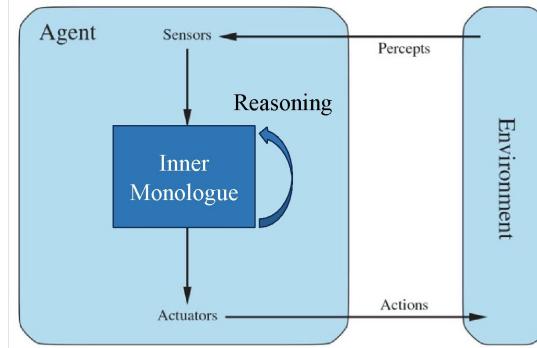
- Agent applications
  - in digital world
  - in physical world
- Agent data
  - via human demonstrations
  - via synthesis and simulation
  - via internet-scale data
- Agent evaluation
  - via benchmarks
  - via LLMs/VLMs
  - via crowdsourcing

# Agent applications

# Agent applications

Agent applications by embodiment

- Digital world
  - Coding agents
  - Gaming agents
  - Mobile agents
  - Web/app agents
  - Computer agents
- Physical world
  - Robotics

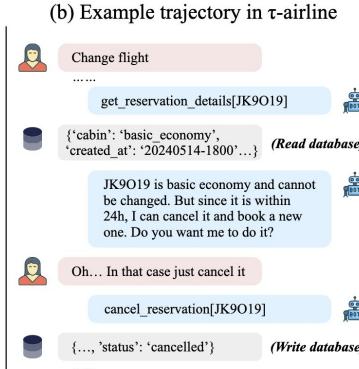
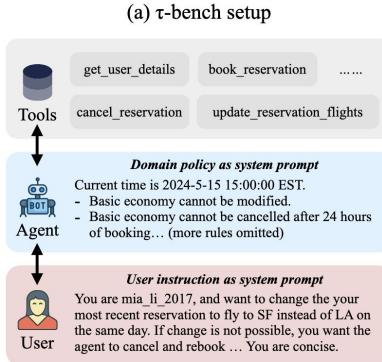


# Agent applications in digital world

# Coding agents

## API/functional calling for tool use

- Environment: software systems such as databases, app/web services...
- Observation space: API docs, system info, error messages and logs...
- Action space: function calls, error handling routines...



{"order\_id": "#W2890441",  
"user\_id": "mei\_davis\_8935",  
"items": [  
    {"name": "Water Bottle",  
    "product\_id": "8310926033",  
    "item\_id": "2366567022",  
    "price": 54.04,  
    "options": {  
        "capacity": "1000ml",  
        "material": "stainless  
                  steel",  
        "color": "blue"  
    }, [...], ...]}

(a) An orders database entry in  $\tau$ -retail.

```
def return_delivered_order_items(  
    order_id: str,  
    item_ids: List[str],  
    payment_method_id: str,  
) -> str: ...  
  
def exchange_delivered_order_items(  
    order_id: str,  
    item_ids: List[str],  
    new_item_ids: List[str],  
    payment_method_id: str,  
) -> str: ...
```

(b) An API tool in  $\tau$ -retail.

## Return delivered order  
- After user confirmation, the order status  
will be changed to 'return requested'...  
  
## Exchange delivered order  
- An order can only be exchanged if its  
status is 'delivered'...

{"instruction": "You are Mei Davis in 80217. You want to return the water bottle, and exchange the pet bed and office chair to the cheapest version. Mention the two things together. If you can only do one of the two things, you prefer to do whatever saves you most money, but you want to know the money you can save in both ways. You are in debt and sad today, but very brief.",  
"actions": [{"

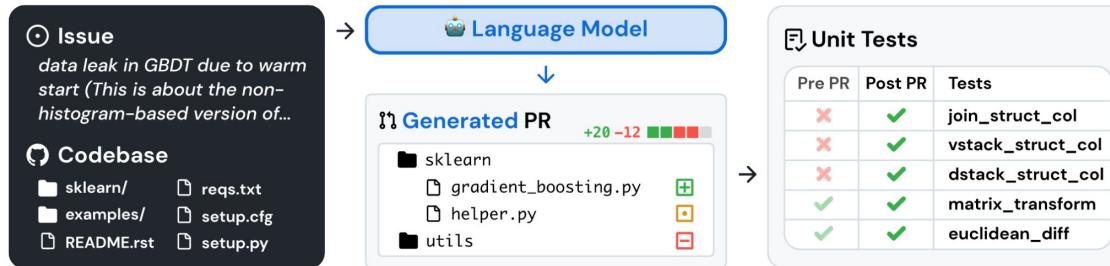
```
    "name": "return_delivered_order_items",  
    "arguments": {  
        "order_id": "#W2890441",  
        "item_ids": ["2366567022"],  
        "payment_method_id":  
                  "credit_card_1061405",  
    }},  
    "outputs": ["54.04", "41.64"]}]
```

(d) User instruction ensures only one possible outcome.

# Coding agents

## Project-level coding tasks

- Environment: project code repos, filesystems, IDEs...
- Observation space: code files, exe outputs, docs, errors, commit history...
- Action space: code edits, file search/view, test updates...



# Agents for software development

Coding agents won't be our focus in this tutorial.

## Agents for Software Development

Graham Neubig



All Hands AI

# Gaming agents

## Digital games

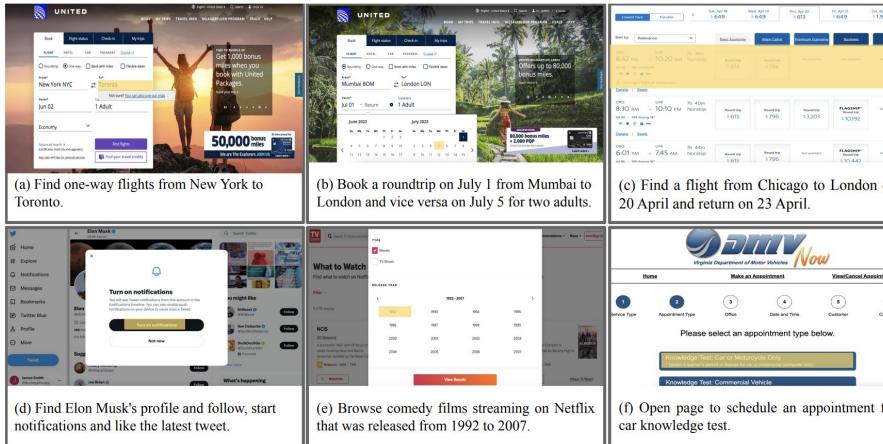
- Environment: game worlds/levels...
- Observation space: screenshots of game states, inventory, location...
- Action space: game controls (e.g., drop, move, attack, resource management...)



# Web/app agents

## Web/app use

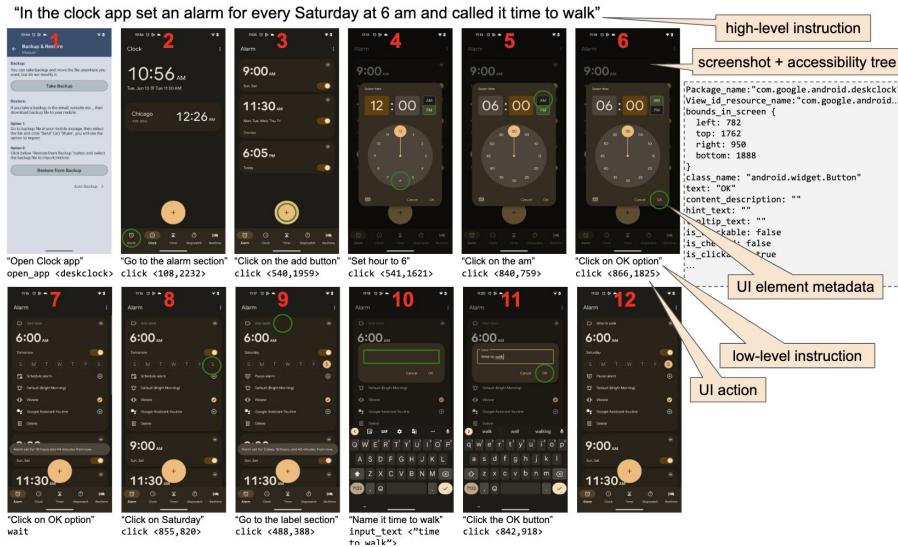
- Environment: web browsers/apps
- Observation space: screenshots, DOM trees, HTML, historical actions...
- Action space: browser/app controls (e.g., click, type, scroll, drag, hover...)



# Mobile agents

## Mobile use

- Environment: mobile device systems
- Observation space: screenshots, a11y trees, HTML, historical actions...
- Action space: mobile controls (e.g., tap, type, swipe...)



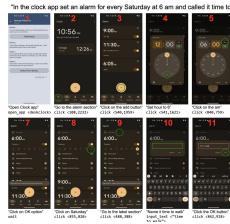
# Universal digital environment



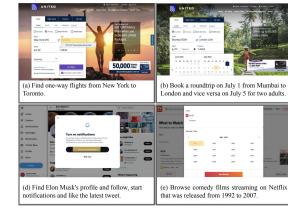
Coding



Gaming

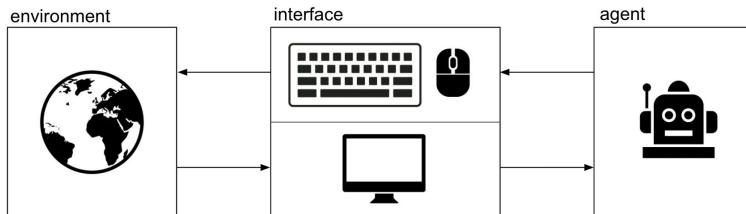


Mobile



Web/apps

Can we study all digital AI agents in a **single** environment with **unified** observation and action spaces?

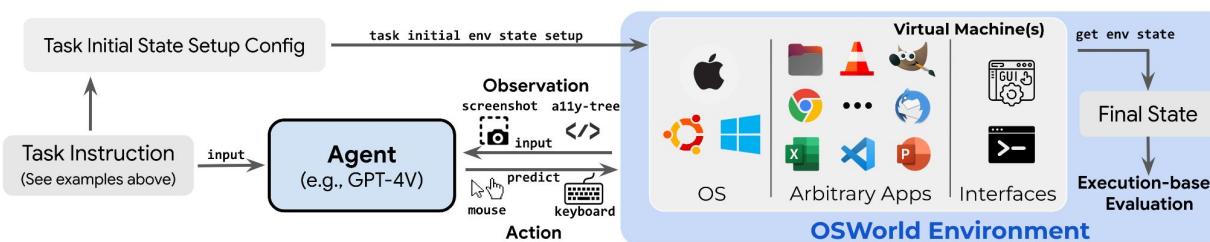


# Computer use agents

Computer use for universal digital tasks

- Environment: desktop operating systems
- Observation space: desktop screenshots, a11y trees, historical actions...
- Action space: keyboard/mouse controls (e.g., click, type, drag, shortcuts)

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



# Computer use agents

## 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]%	<b>75.00%</b>
Office	7.7%	[2.9, 12.5]%	17.9%	[11.0, 24.8]%	<b>71.79%</b>
Daily	16.7%	[8.4, 25.0]%	24.4%	[14.9, 33.9]%	<b>70.51%</b>
Professional	24.5%	[12.5, 36.5]%	40.8%	[27.0, 54.6]%	<b>73.47%</b>
Workflow	7.9%	[2.6, 13.2]%	10.9%	[4.9, 17.0]%	<b>73.27%</b>
Overall	14.9%	[11.3, 18.5]%	22%	[17.8, 26.2]%	<b>72.36%</b>

Anthropic recent computer use agent results of OSWorld

Agent applications  
in physical world

# Robotic agents

Robotics for physical interaction

- Environment: physical world spaces
- Observation space: visual input, sensor readings, physical states, proprioception...
- Action space: motor controls (e.g., move, grasp, manipulate...)



# Agent application overview

## Physical agent tasks

- Observation: sensor data streams
- Control complexity: high



- Task distribution: more concentrated, natural



- Data collection: very hard (simulation)
- Evaluation: very hard (simulation)
- Deployment: complex (sim2real gap)

## Digital agent tasks

- Observation: screen/UI states
- Control complexity: Low



- Task distribution: long tail, reasoning-intensive

A screenshot of a GitHub pull request interface. At the top left is an "Issue" card with the title "data leak in GBDT due to warn start (This is about the non-histogram-based version of...)" and a "Codebase" section showing files like "sklearn/", "examples/", "README.rst", and "setup.py". To the right is a "Language Model" card. Below is a "Generated PR" card with a summary "+28 -12" and a list of files: "sklearn/gradient\_boosting.py", "helper.py", and "utils".

Coding



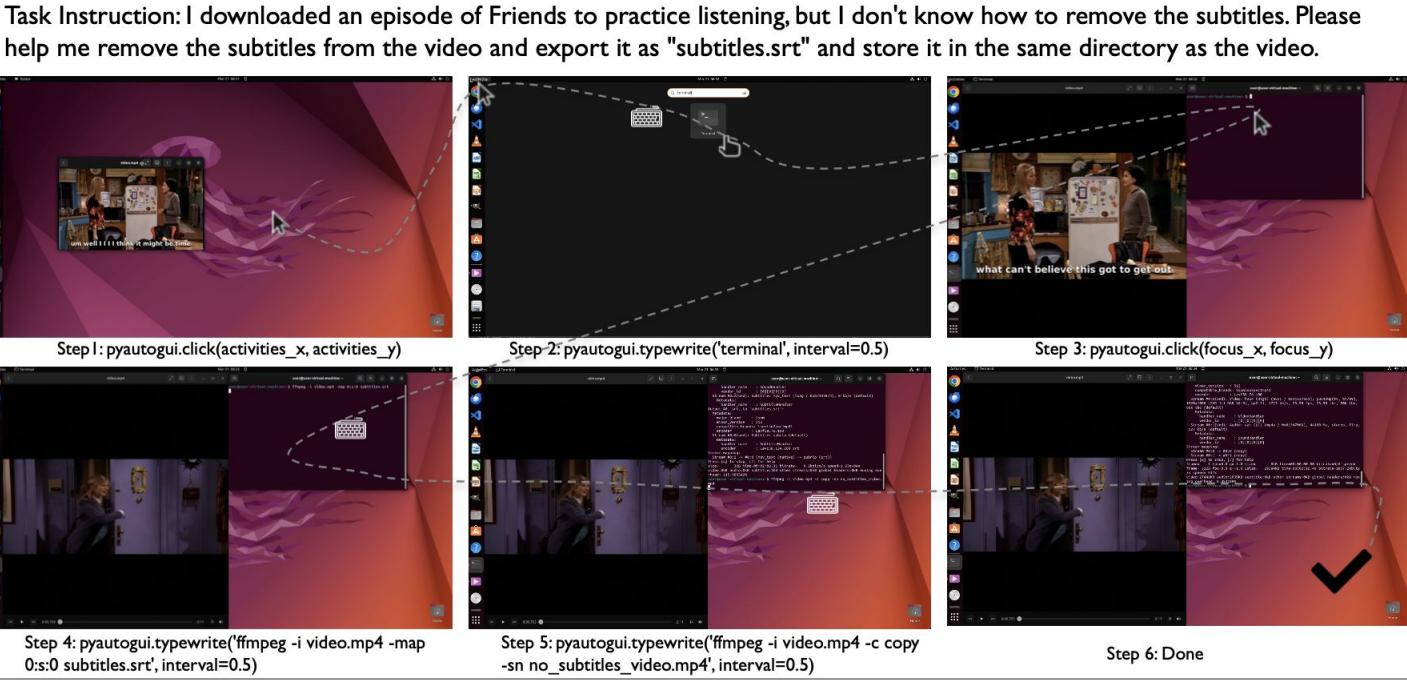
Professional workflows

- Data collection: hard (real env)
- Evaluation: hard (real env)
- Deployment: easy (no sim2real gap)

# Agent data

# Agent data example

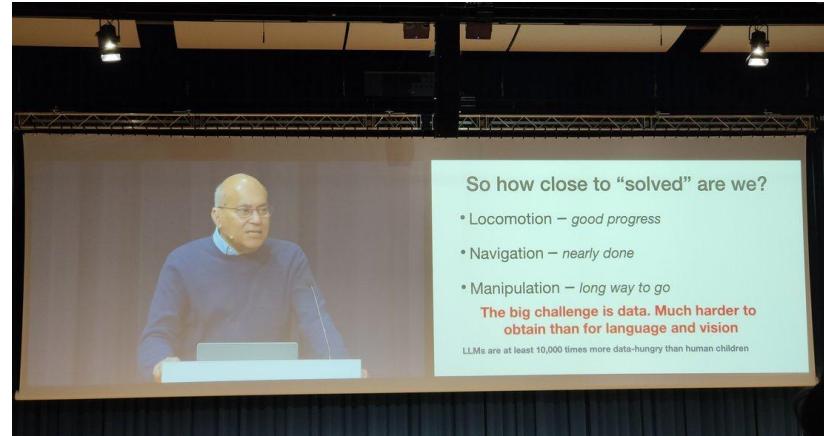
Task goal aligned trajectories (observation-action pairs)



# Agent data: a big challenge!

Agent data is hard to get directly from internet-scale text and videos due to **embodiment**.

- Complex data collection infrastructure
- Complex observation-action interaction in diverse environments
- Goal aligned trajectories



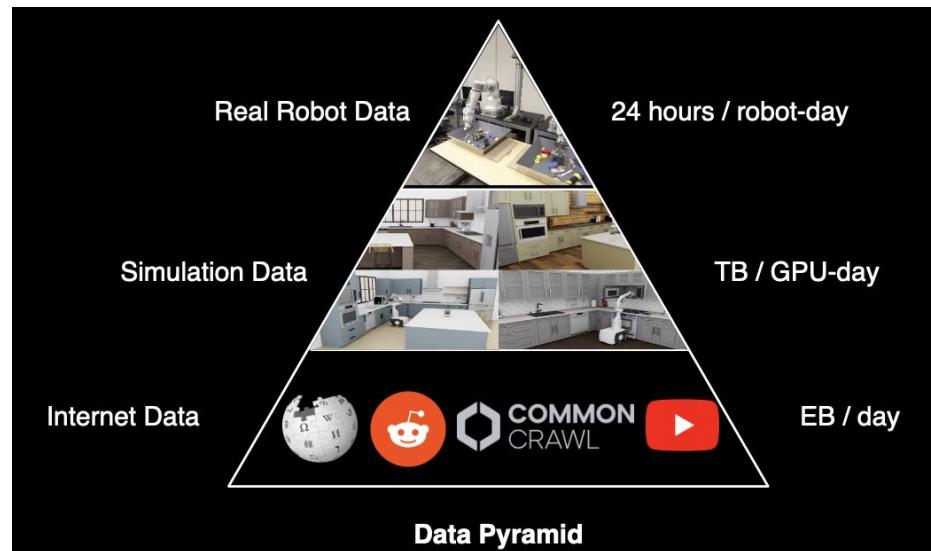
Jitendra's talk @ CoRL 2024



# Scaling agent data

Scaling agent data through

- Human demonstrations
- Synthesis/simulation
- Internet-scale data



Agent data  
via human demonstrations

# Agent data challenge 1: hard to collect

Human demonstration pipeline

- Task definition
- Infrastructure setup
- Task initial environment config
- Human demonstration recording
- Data verification



# Agent data: not yet at scale

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

Existing digital agent data

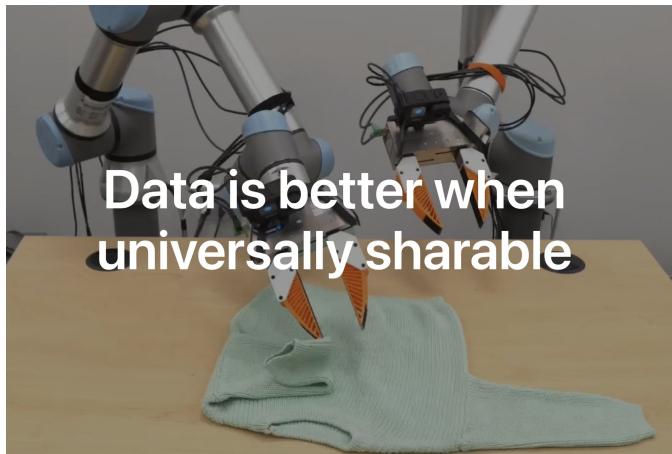
Feature	RoboCasa	AI2-THOR	Habitat 2.0	iGibson 2.0	RLBench	Behavior-1K	robomimic	ManSkill 2	OPTIMUS	LIBERO	MimicGen
Mobile Manipulation	✓	✓	✓	✓	✗	✓	✗	✓	✗	✗	✓
Room-Scale Scenes	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗	✓
Realistic Object Physics	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
AI-generated Tasks	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
AI-generated Assets	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Photorealism	✓	✓	✓	✗	✗	✓	✗	✓	✓	✗	✗
Cross-Embodiment	✓	✓	✗	✓	✓	✓	✗	✗	✗	✗	✓
Num Tasks	100	-	3	6	100	1000	8	20	10	130	12
Num Scenes	120	-	1	15	1	50	3	-	4	20	1
Num Object Categories	153	-	46	-	28	1265	-	-	-	x	-
Num Objects	2509	3578	169	1217	28	5215	15	2144	72	x	40
Human Data	✓	✗	✗	✓	✗	✗	✓	✗	✗	✓	✓
Machine-Generated Data	✓	✗	✗	✗	✓	✗	✓	✓	✓	✗	✓
Num Trajectories	100K+	-	-	-	-	0	6K	30K	245K	5K	50K

Existing robotic agent data

# Agent data challenge 2: hard to share

Heterogeneous agent data formats

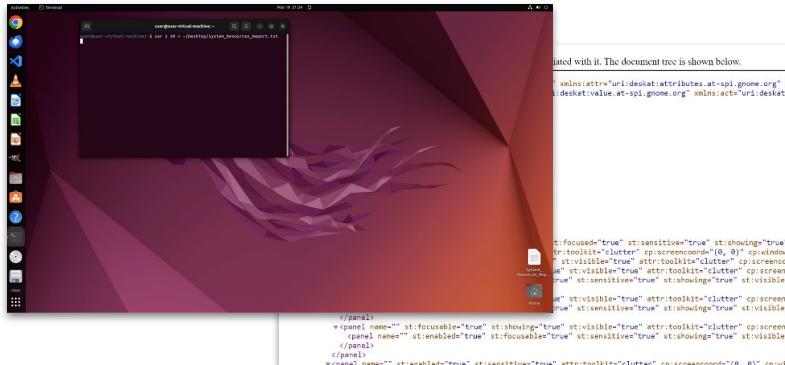
- Data from various platforms and embodied environments produces different observation and action spaces
- Make data merging and standardization difficult, hindering development of general-purpose agents



# Unifying digital agent data

## Digital agent data unification

- Observation: screenshots (or a11y trees, but not reliable)
- Actions: universal computer mouse and keyboard controls



Observation: screenshots/a11y trees

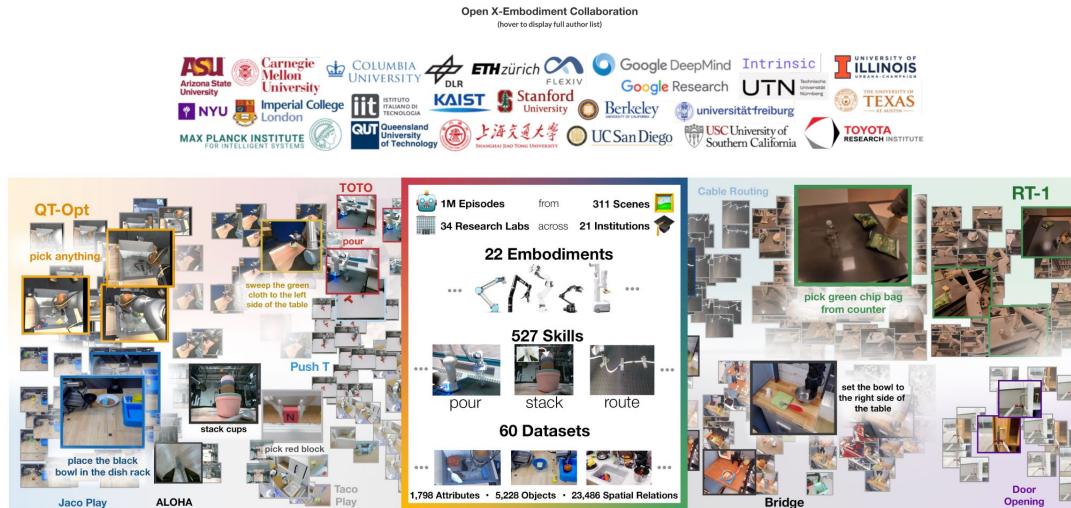
Category	Action Space
Basic Actions	pyautogui.moveTo(x, y) pyautogui.click(x, y) pyautogui.write('text') pyautogui.press('enter') pyautogui.hotkey('ctrl', 'c') pyautogui.scroll(200) pyautogui.dragTo(x, y)
Pluggable Actions	browser.select_option(x, y, value) mobile.swipe(from, to) mobile.home() mobile.back() mobile.open_app(name) terminate(status) answer(text)
	...

Actions: pyautogui computer control actions  
with pluggable actions

# Unifying robotic agent data

Unifying data from diverse robotic platforms and sensors to enable large-scale agent training.

## Open X-Embodiment: Robotic Learning Datasets and RT-X Models



# Unifying robotic agent data collection infrastructure

Simplifying and unifying robotic data collection hardwares



## Universal Manipulation Interface

In-The-Wild Robot Teaching Without In-The-Wild Robots



Human Demonstration

in Any Environment  
(visual diversity)



for Any Actions  
(action diversity)

Dynamic

Precise

for Many Robots  
(embodiment diversity)

6DoF

6DoF

7DoF

7DoF

## OmniH2O: Universal and Dexterous Human-to-Humanoid Whole-Body Teleoperation and Learning

Tairan He\* Zhengyi Luo\* Xialin He\* Wenli Xiao Chong Zhang Weinan Zhang Kris Kitani Changliu Liu Guanya Shi

Carnegie Mellon University SHANGHAI JIAO TONG UNIVERSITY

CoRL 2024



(a) Teleoperation

(b) Autonomous Agent

# Unifying digital agent data collection infrastructure

Unifying digital agent data collection with our AgentNet tool

- Universal platform for digital agent data collection and verification in a unified data format

AgentNet Documentations

AgentNet Annotation Tool

AgentNet annotation tool is an annotation app that collects various types of computer data (actions such as clicks and scrolls, desktop recordings and webpage HTML etc.) while you work on your computer tasks.

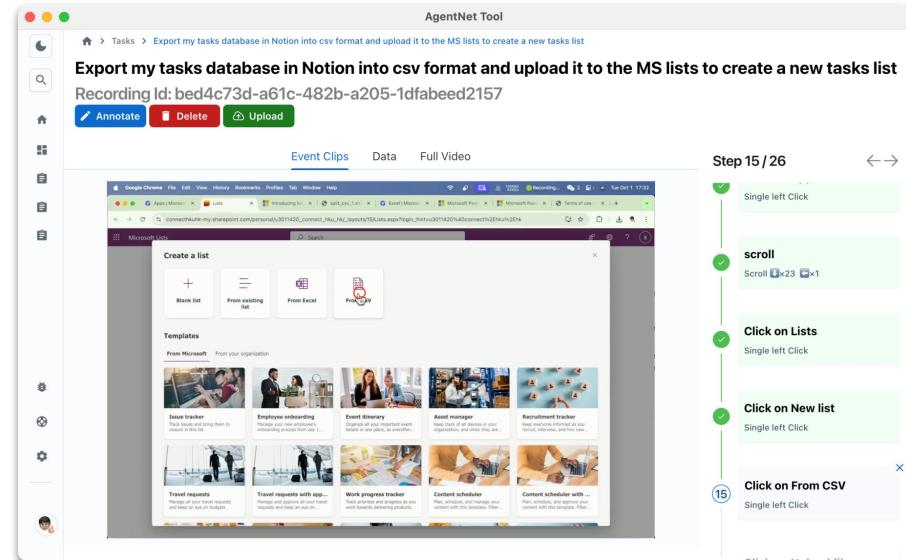
In order to use AgentNet tool to annotate task examples, you need to first install and setup some tools (Part 1) and then follow the annotation guideline (Part 2) to annotate qualified task examples.

- Part 1: Installation: Installation and setup for MacOS or Windows.
- Part 2: Annotation Guidance: Annotation pipeline and requirements.
- Part 3: FAQ (Optional): Frequently Asked Questions and common bugs solutions, for MacOS or Windows

**Commercial Use Prohibited**

We would like to clarify that our application is currently not fully public and is still in a pre-release phase. At this time, it is intended for internal testing and academic purposes only.

Commercial use of AgentNet Tool without our permission is strictly prohibited, and any such use may result in legal action. If your organization is interested in discussing the usage of our tool, please reach out to us via email: [xywang26@gmail.com](mailto:xywang26@gmail.com) or [salyp@mit.edu](mailto:salyp@mit.edu).



# **Human demonstration is hard to scale**

Challenges in scaling human demonstration data collection

- Expensive and complex infrastructure setup
- Expert time & cost
- Task coverage

Possible solutions

- Data synthesis or simulation
- Leveraging internet-scale data

Agent data  
via synthesis and simulation

# Synthesizing digital agent data

Converting online tutorials into direct training demonstrations, making human-oriented instruction materials usable for training AI systems

## Synatra: Turning Indirect Knowledge into Direct Demonstrations for Digital Agents at Scale

Tianyue Ou<sup>1</sup>, Frank F. Xu<sup>1</sup>, Aman Madaan<sup>1</sup>,

Jianui Liu<sup>1</sup>, Robert Lo<sup>1</sup>, Abishek Sridhar<sup>1</sup>,

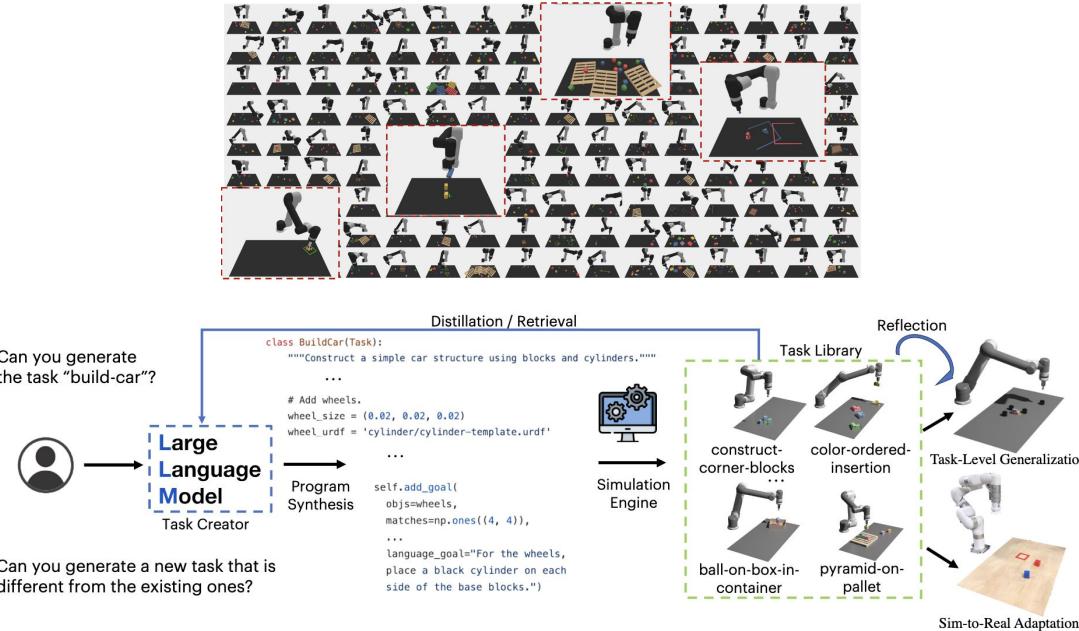
Sudipta Sengupta<sup>2</sup>, Dan Roth<sup>2</sup>, Graham Neubig<sup>1</sup>, Shuyan Zhou<sup>1</sup>

<sup>1</sup>Carnegie Mellon University, <sup>2</sup>Amazon AWS AI



# Scaling robotic data via simulation

Generating simulation environments and expert demonstrations by leveraging LLM's grounding and coding ability.

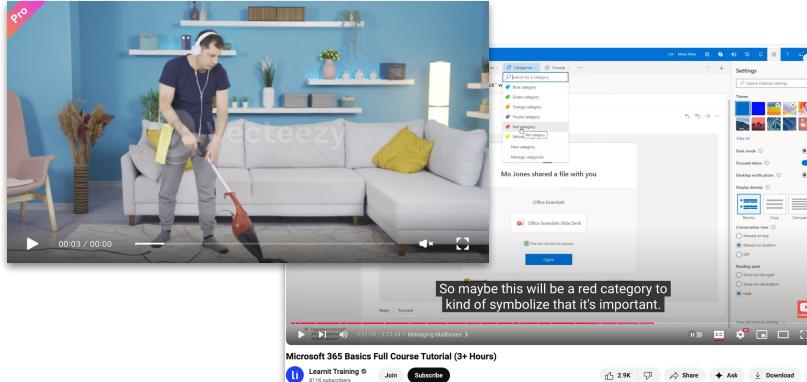


# Synthesizing good agent data is still challenging

Challenges in agent data synthesis and simulation

- Limited foundation model capabilities
- World knowledge or exploration limitations
- Sim2real gap

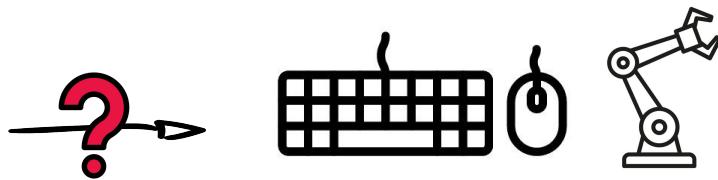
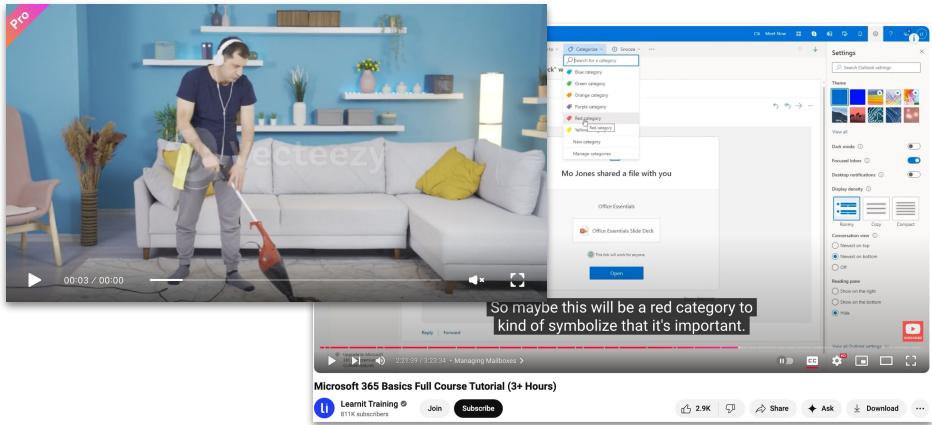
Possible solution: leveraging internet-scale human demonstration video data?



Agent data  
via internet-scale data

# Internet-scale data for agent learning

Numerous videos exist online showing humans demonstrating how to perform agent tasks, ***but without grounded trajectories!***



Control actions

# Digital agent learning from online videos

## Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos

Bowen Baker<sup>\*†</sup> Ilge Akkaya<sup>\*†</sup> Peter Zhokhov<sup>\*†</sup> Joost Huizinga<sup>\*†</sup>  
bowen@openai.com ilge@openai.com peterz@openai.com joost@openai.com

Jie Tang<sup>\*†</sup> Adrien Ecoffet<sup>\*†</sup> Brandon Houghton<sup>\*†</sup> Raul Sampedro<sup>\*†</sup>  
jetang@openai.com adrien@openai.com brandon@openai.com raulsamt@gmail.com

Jeff Clune<sup>††</sup>  
jclune@gmail.com

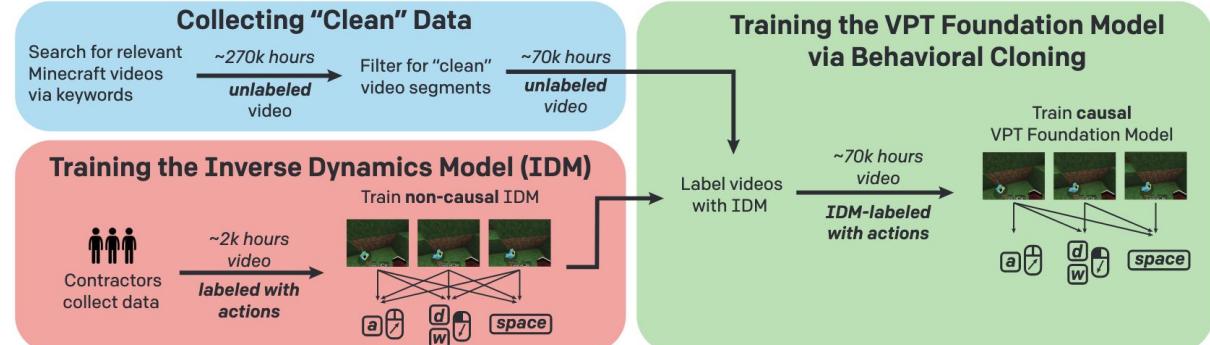


Figure 2: Video Pretraining (VPT) Method Overview.

# Robotic learning from internet videos

## LAPA: Latent Action Pretraining from Videos

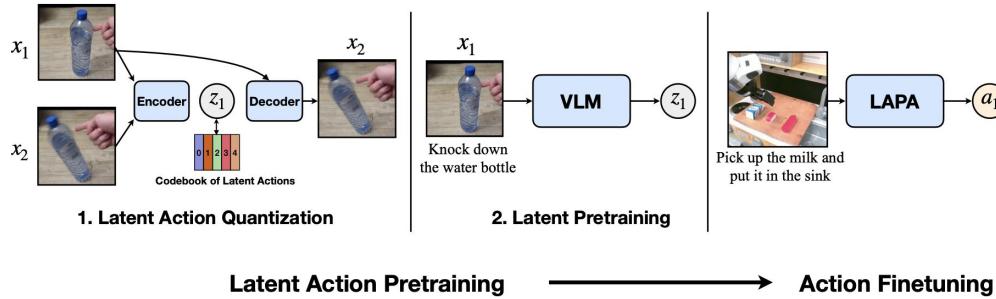
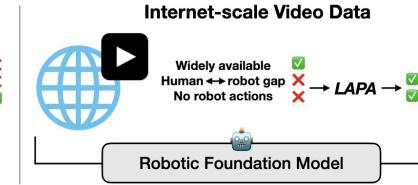
Seonghyeon Ye<sup>\*</sup>, Joel Jang<sup>\*\*</sup>,  
Byeongguk Jeon<sup>1</sup>, Sejune Joo<sup>1</sup>, Jianwei Yang<sup>3</sup>, Baolin Peng<sup>3</sup>, Ajay Mandlekar<sup>4</sup>,  
Reuben Tan<sup>3</sup>, Yu-Wei Chao<sup>4</sup>, Yuchen Lin<sup>5</sup>, Lars Liden<sup>3</sup>,  
Kimin Lee<sup>††</sup>, Jianfeng Gao<sup>3†</sup>, Luke Zettlemoyer<sup>2†</sup>, Dieter Fox<sup>2,4†</sup>, Minjoon Seo<sup>1†</sup>

<sup>1</sup>KAIST <sup>2</sup>University of Washington  
<sup>3</sup>Microsoft Research <sup>4</sup>NVIDIA <sup>5</sup>Allen Institute for AI  
\* Equal contribution, † Equal advising

### Large-Scale Robot Datasets



Expensive to collect  
Requires robot hardware  
Contains robot actions



# **Internet-scale data is not perfect**

Challenges in using internet data for agent training

- Missing grounded action sequences, environmental state info
- Observation-action alignment
- Unclear task objectives from video alone

# ImageNet in agent learning?



# Agent evaluation

# Evaluation in the era of LLMs is hard



Andrej Karpathy

@karpathy

...

Nice, a serious contender to [@lmsysorg](#) in evaluating LLMs has entered the chat.

LLM evals are improving, but not so long ago their state was very bleak, with qualitative experience very often disagreeing with quantitative rankings.

This is because good evals are very difficult to build - at Tesla I probably spent 1/3 of my time on data, 1/3 on evals, and 1/3 on everything else. They have to be comprehensive, representative, of high quality, and measure gradient signal (i.e. not too easy, not too hard), and there are a lot of details to think through and get right before your qualitative and quantitative assessments line up. My goto pointer for some of the fun subtleties is probably the Open LLM Leaderboard MMLU writeup:  
[github.com/huggingface/bl...](https://github.com/huggingface/bl...)

# **Agent evaluation is even more challenging...**

Challenges in agent evaluation

- Real-world environmental setup complexity
- Task coverage
- Open-ended success criteria
  - Multiple valid solution paths
  - Cannot script evaluation metrics, need for human judgment
- Evaluation beyond task success

Agent evaluation

- via benchmarks
- via LLMs/VLMs
- via crowdsourcing

Agent evaluation  
via benchmarks

# **How to define good agent benchmarks?**

- Natural and challenging tasks

# How to define good agent benchmarks?

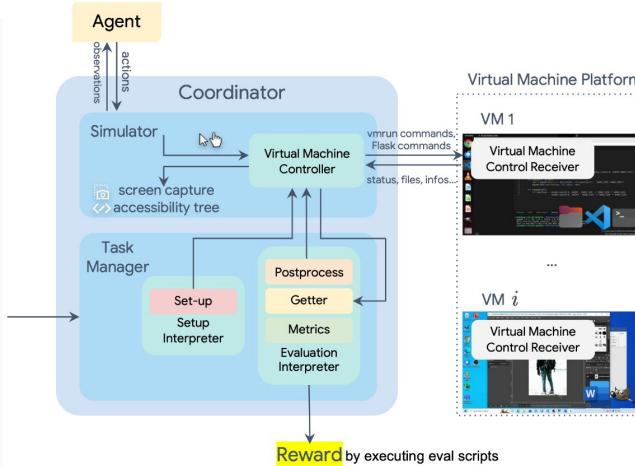
- Natural and challenging tasks
- Good agent evaluation framework
  - Realistic agent environment
  - Automatic initial task state setup
  - Automatic task evaluation: execution-based scripts to compare final states

```
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=xxxxx"},

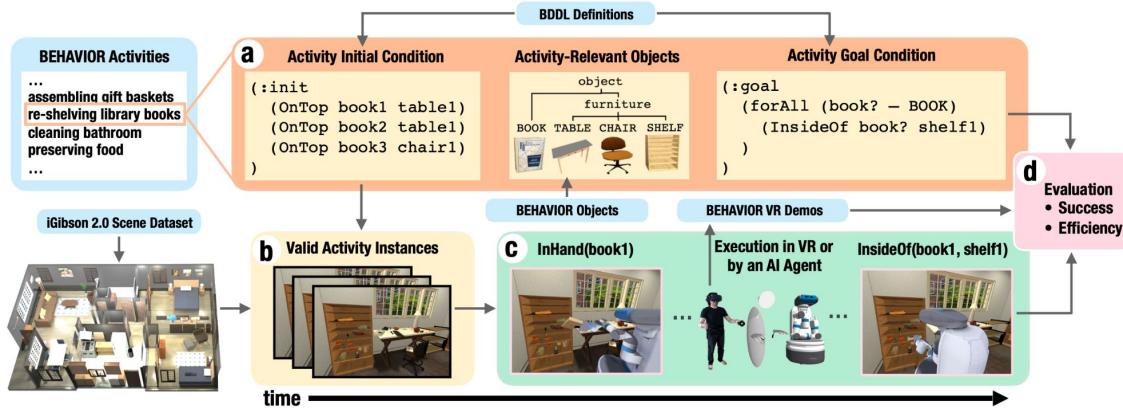
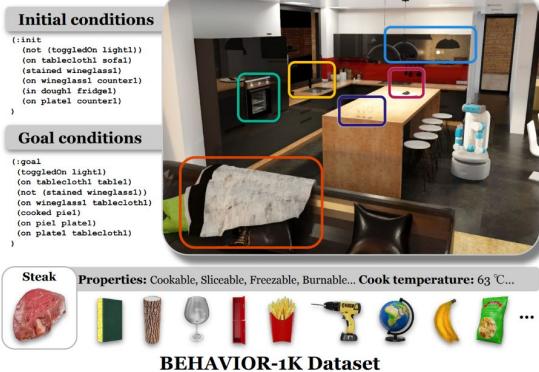
      {"path": "/home/user/Desktop/receipt_0.jpeg",
       "url": "https://drive.google.com/uc?id=xxxxx"},...]}],
    "type": "open",
    "parameters": {"path": "/home/user/Desktop/my_bookkeeping.xlsx"}],
  "evaluator": {"postconfig": [{"type": "activate_window",
    "parameters": {"window_name": "my_bookkeeping.xlsx - 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",... ]}]}]}]}]
```



# Robotic task evaluation



# More agent evaluation metrics

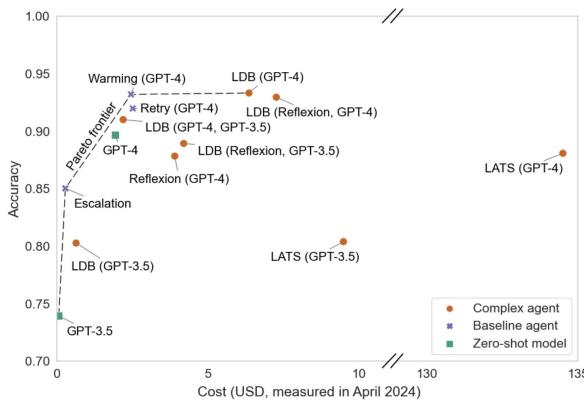
## Other agent evaluation metrics

- Latency - efficiency
  - Compute aware success rate
  - Real time evaluation
- Robustness
  - Generalization to unseen domains, tasks, apps

### AI Agents That Matter

Sayash Kapoor\*, Benedikt Stroebel\*, Zachary S. Siegel, Nitya Nadgir, Arvind Narayanan

Princeton University  
July 2, 2024



# More agent evaluation metrics

Other agent evaluation metrics

- Safety - will be covered by Diyi



# Limitations of agent benchmarks

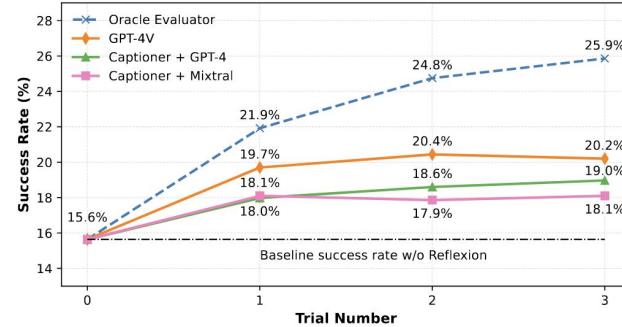
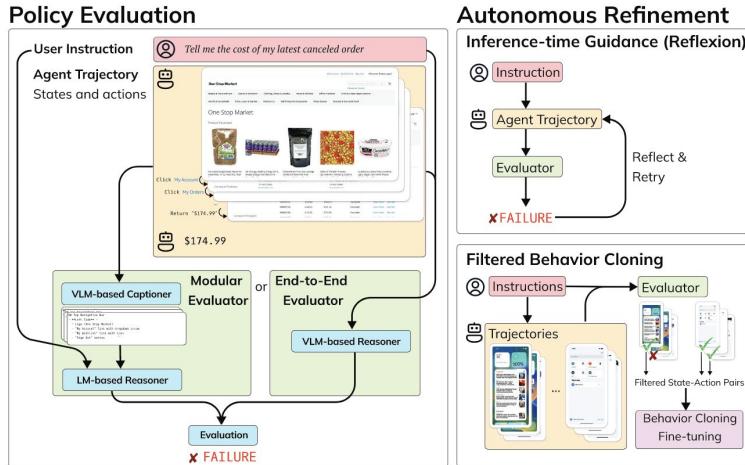
- Only can write evaluation scripts for very limited tasks, time-consuming
- Cannot script evaluation metrics for open-answer tasks

Possible solution: leveraging LLM/VLMs to automatically evaluate agent tasks?

Agent evaluation  
via LLMs/VLMs

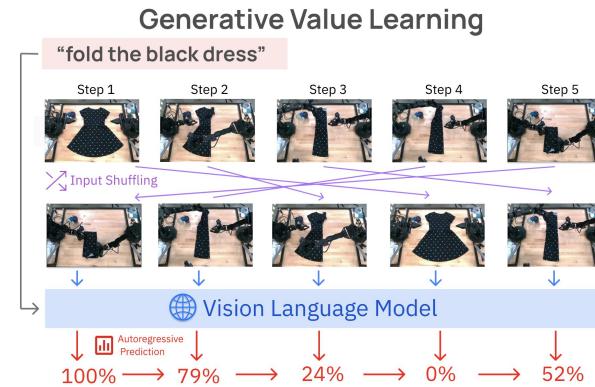
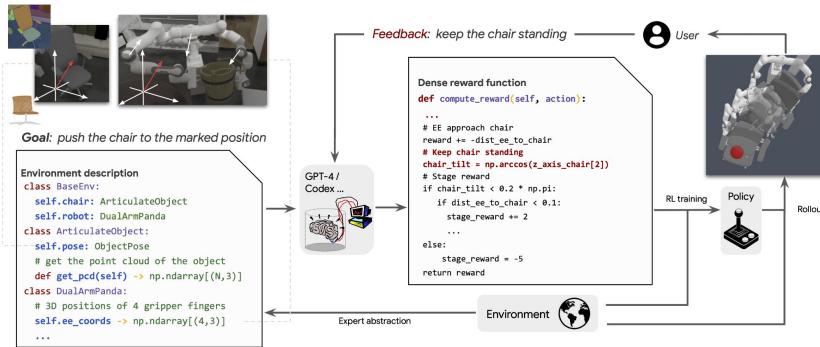
# Automatic agent evaluation

Automatically evaluate user instructions and arbitrary agent trajectories with LLM/VLMs



# Automatic agent evaluation

- Leveraging coding ability of LLMs to automatically generate reward functions
- Leveraging the world knowledge embedded in VLMs to evaluate task progress



# **Limitations of automatic agent evaluation**

- Limited foundation model capabilities
- Missing personalized task evaluation

Possible solution: how about evaluating agent tasks via crowdsourcing from real users?

- Personalized and robust success criteria capture
- Diverse task scenarios and environments
- Natural interaction and feedback loops
- Hard to overfit

Agent evaluation  
via crowdsourcing

# Chatbot arena

Chatbot Arena is not embodied for agent evaluation

The screenshot shows the Chatbot Arena LLM Leaderboard interface. At the top, there's a header with the title 'Chatbot Arena LLM Leaderboard' and a subtext: 'Backed by over 1,000,000+ community votes, our platform ranks the best LLM and AI chatbots. Explore the top AI models on our LLM leaderboard!' Below this is a 'Chat now!' button.

Two AI models are displayed side-by-side:

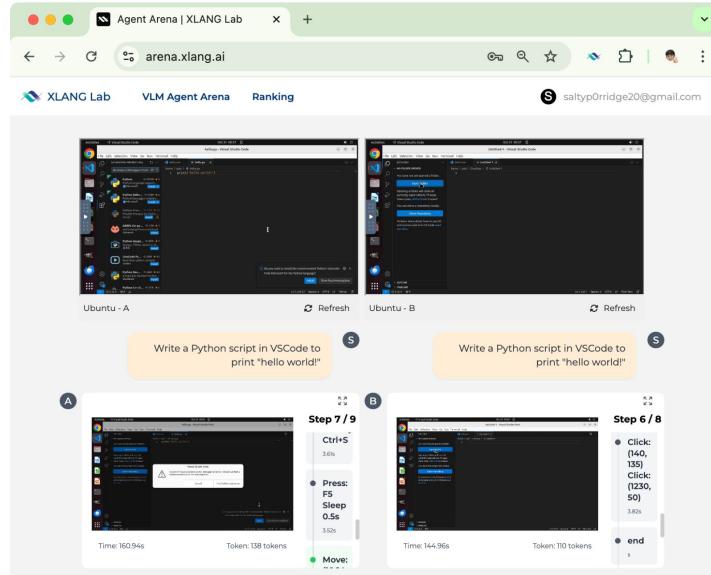
- Model A:** The response is a link titled 'Defining Language Agents: A Comprehensive Overview'.
- Model B:** The response is a detailed text block explaining what language agents are, their core function, key characteristics, and components.

Below the responses, there are four buttons for user preference: 'A is better', 'B is better', 'Tie', and 'Both are bad'. There's also a text input field for entering a prompt and a 'Send' button. At the bottom, there are buttons for 'Random Image', 'New Round', 'Regenerate', and 'Share'.

# Agent arena for digital agent task evaluation

Computer Agent Arena: <https://arena.xlang.ai>

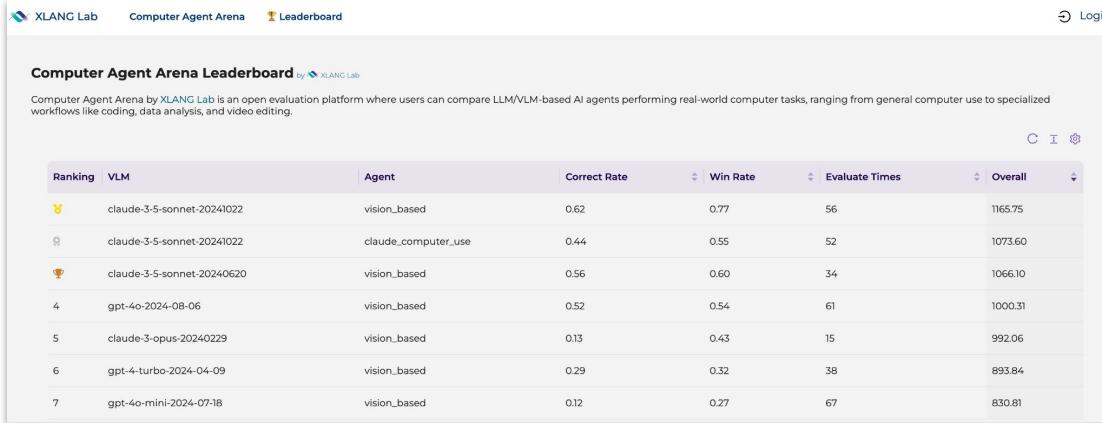
- an open evaluation platform where users can compare LLM/VLM-based AI agents performing real-world computer tasks, ranging from general computer use to specialized workflows like coding, data analysis, and video editing



# Current agent arena leaderboard

Computer Agent Arena: <https://arena.xlang.ai>

- More advanced AI agents are expected to emerge in the near future!



The screenshot shows the 'Computer Agent Arena Leaderboard' page. The page has a header with the XLANG Lab logo, 'Computer Agent Arena', 'Leaderboard', and a 'Login' button. Below the header is a brief description of the platform: 'Computer Agent Arena by XLANG Lab is an open evaluation platform where users can compare LLM/VLM-based AI agents performing real-world computer tasks, ranging from general computer use to specialized workflows like coding, data analysis, and video editing.' The main content is a table with the following data:

Ranking	VLM	Agent	Correct Rate	Win Rate	Evaluate Times	Overall
1	claudie-3-5-sonnet-2024022	vision_based	0.62	0.77	56	1165.75
2	claudie-3-5-sonnet-2024022	claude_computer_use	0.44	0.55	52	1073.60
3	claudie-3-5-sonnet-20240620	vision_based	0.56	0.60	34	1066.10
4	gpt-4o-2024-08-06	vision_based	0.52	0.54	61	1000.31
5	claudie-3-opus-20240229	vision_based	0.13	0.43	15	992.06
6	gpt-4-turbo-2024-04-09	vision_based	0.29	0.32	38	893.84
7	gpt-4o-mini-2024-07-18	vision_based	0.12	0.27	67	830.81