香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

COMP 4901B
Large Language Models

Language Agents and Tools

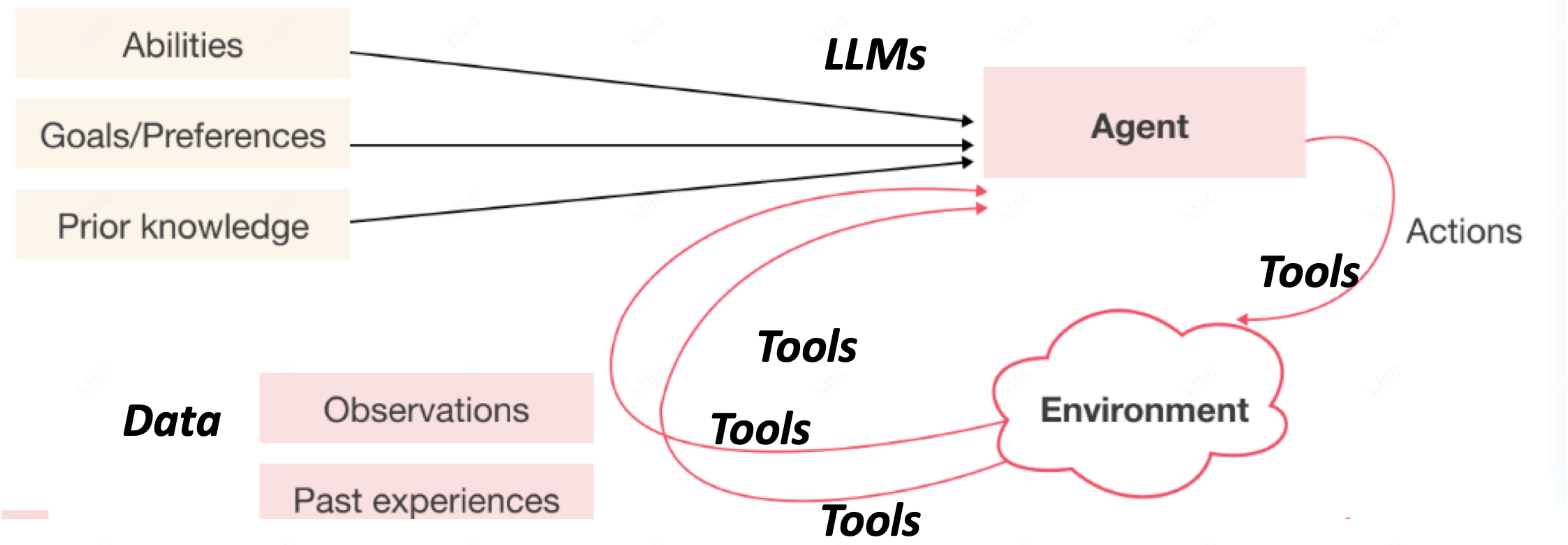
Junxian He

Nov 12, 2025

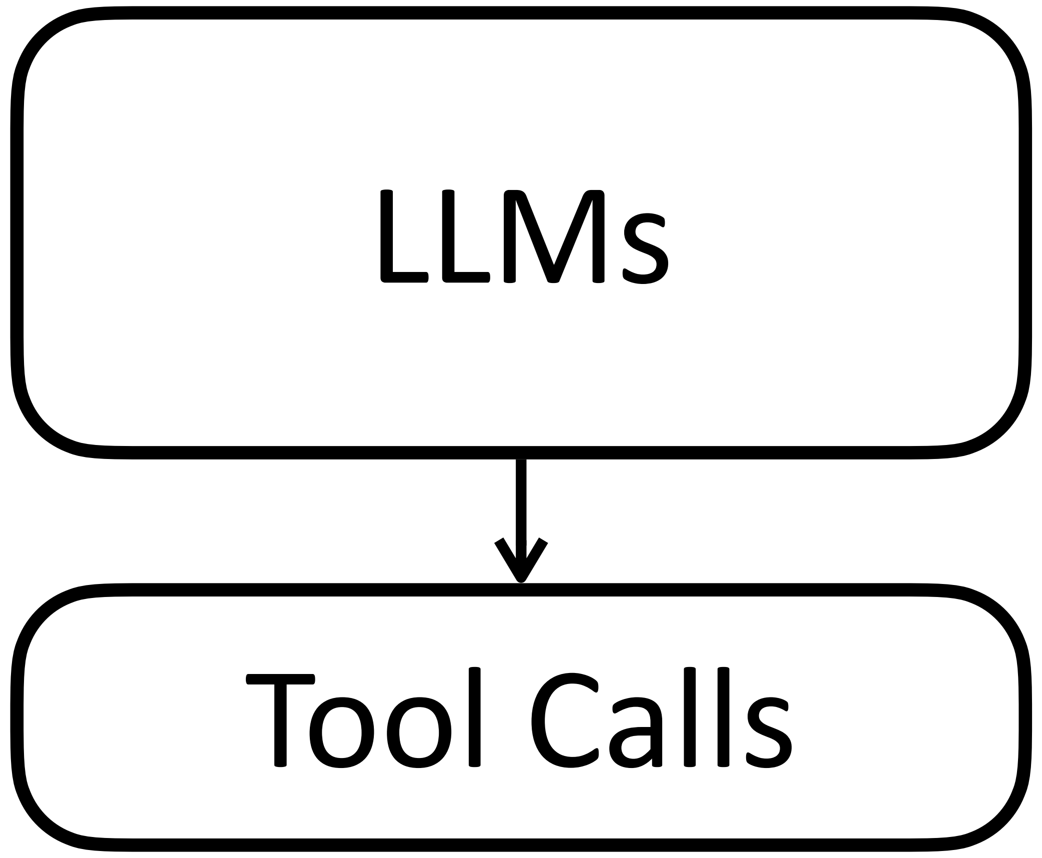
Part of slides are adapted from CMU 11711

Recap: What are Agents

Anything that can be viewed as **perceiving** its environment through sensors and **acting** upon that environment through actuators.

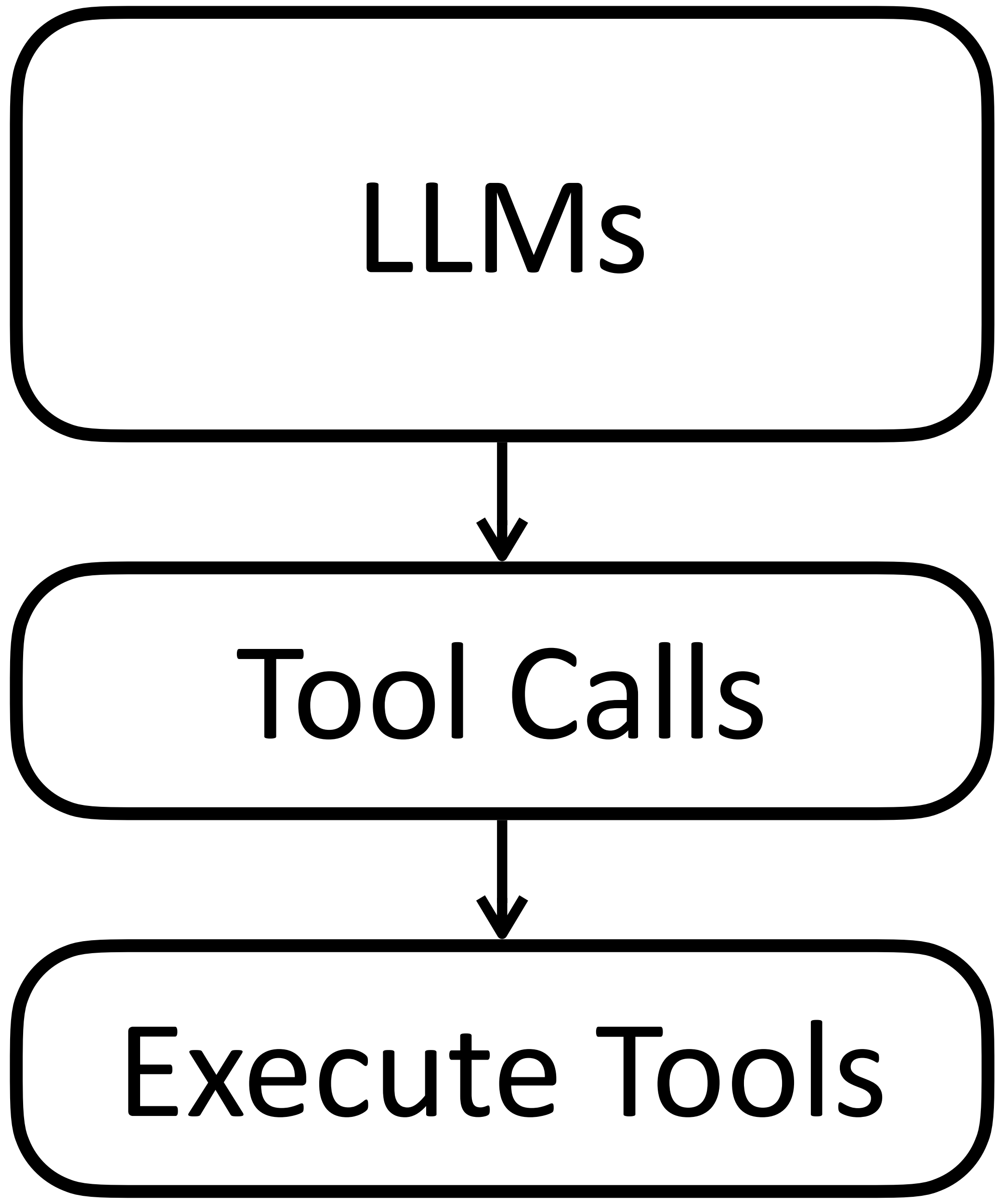


One-Step Tool Call -> Agents

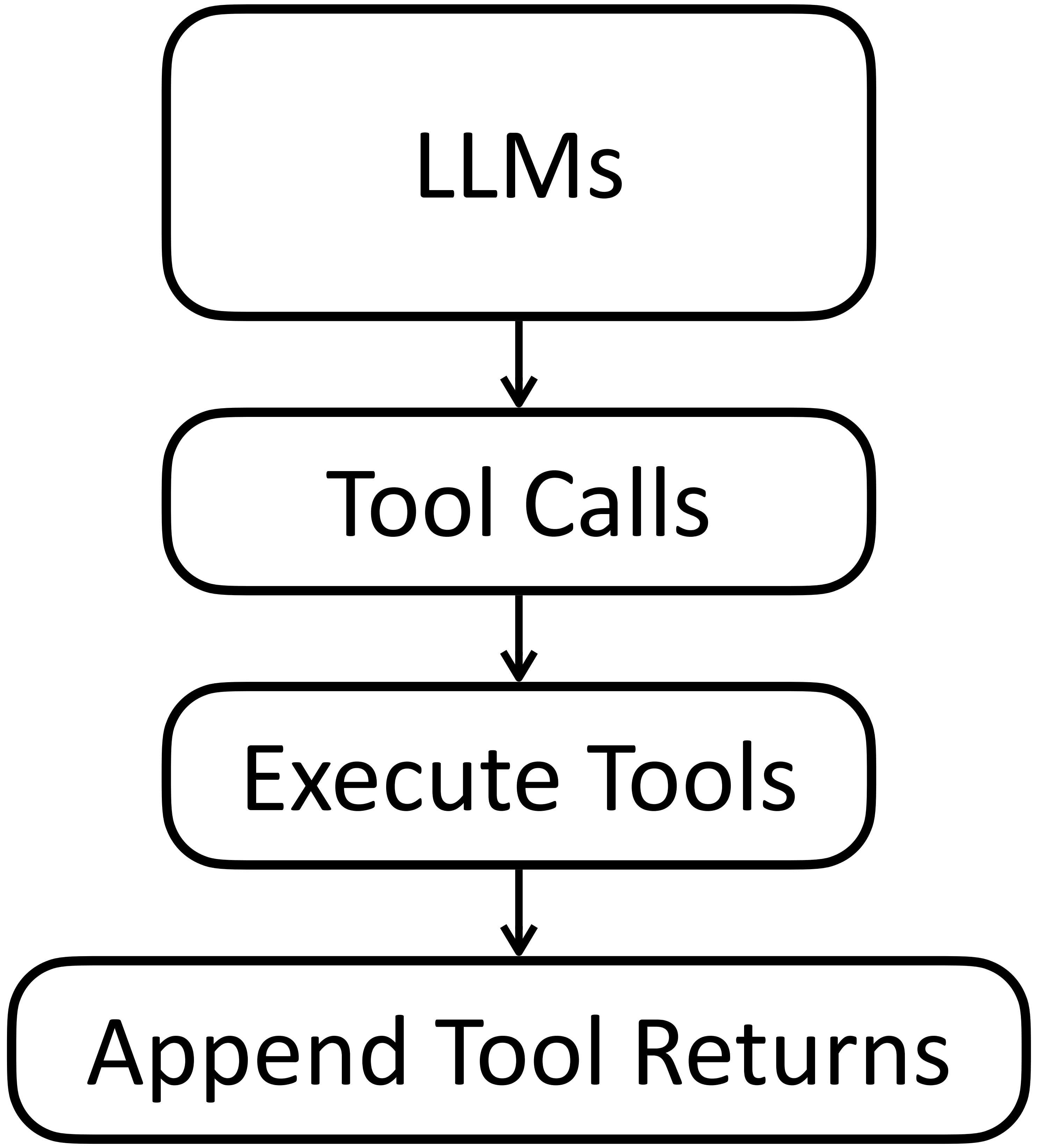



```
{
  "response": "Sure, I'll check the current weather for you.",
  "reasoning": "I need real-time conditions so the user's route
recommendation is accurate.",
  "tool_calls": [
    {
      "name": "get_weather",
      "arguments": {
        "location": "San Jose, CA, US",
        "date": "2025-11-07"
      }
    }
  ],
}
```

One-Step Tool Call -> Agents

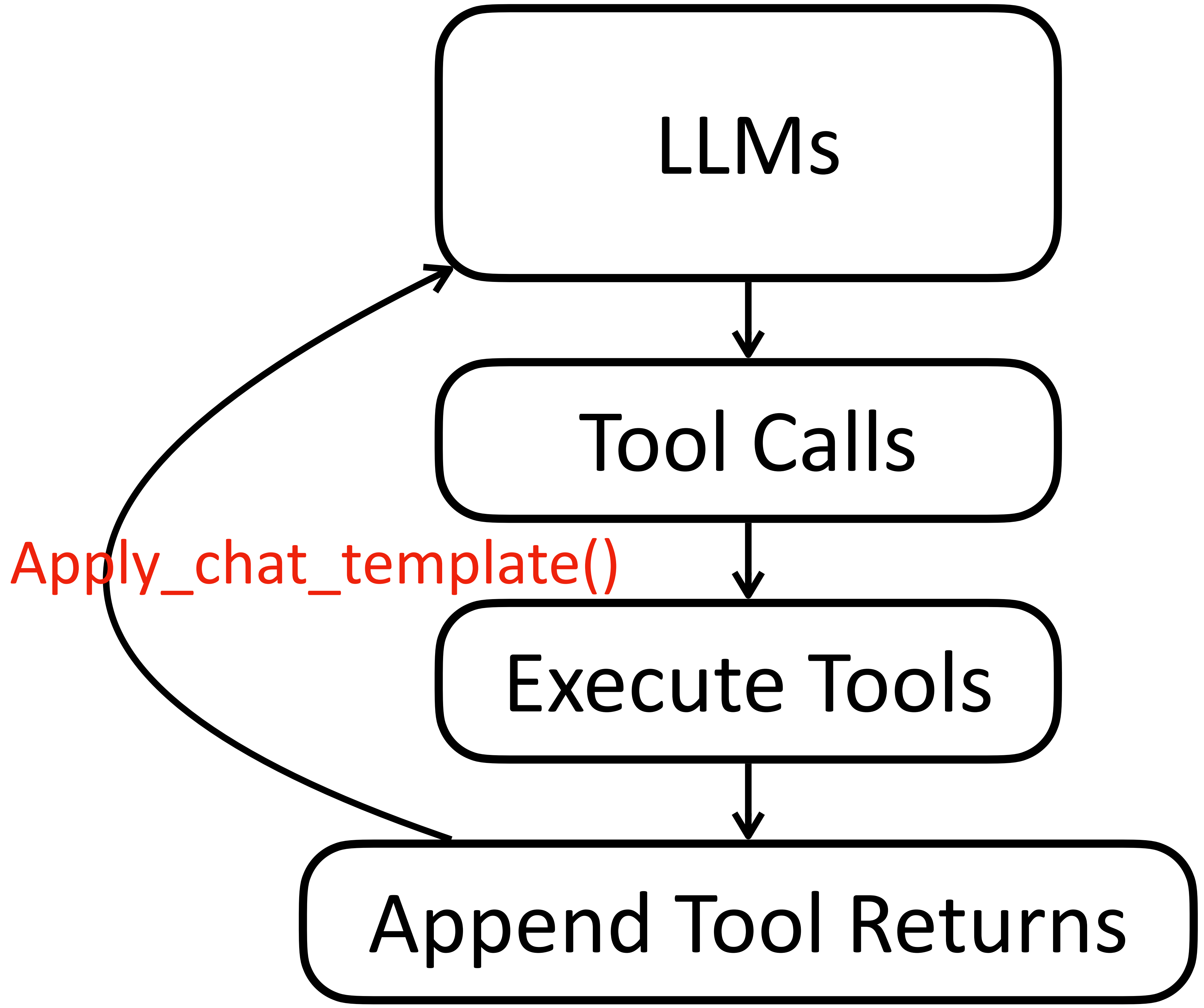


One-Step Tool Call -> Agents




```
{
  "response": "Sure, I'll check the current weather for you.",
  "reasoning": "I need real-time conditions so the user's route recommendation is accurate.",
  "tool_calls": [
    {
      "name": "get_weather",
      "arguments": {
        "location": "San Jose, CA, US",
        "date": "2025-11-07"
      }
    }
  ],
  "tool_return": {
    "temperature": 21.5,
    "condition": "clear",
    "humidity": 60,
    "wind_speed": 10,
    "location": "San Jose, CA, US",
    "date": "2025-11-07"
  }
}
```

One-Step Tool Call -> Agents




```
{
  "response": "Sure, I'll check the current weather for you.",
  "reasoning": "I need real-time conditions so the user's route recommendation is accurate.",
  "tool_calls": [
    {
      "name": "get_weather",
      "arguments": {
        "location": "San Jose, CA, US",
        "date": "2025-11-07"
      }
    }
  ],
  "tool_return": {
    "temperature": 21.5,
    "condition": "clear",
    "humidity": 60,
    "wind_speed": 10,
    "location": "San Jose, CA, US",
    "date": "2025-11-07"
  }
}
```

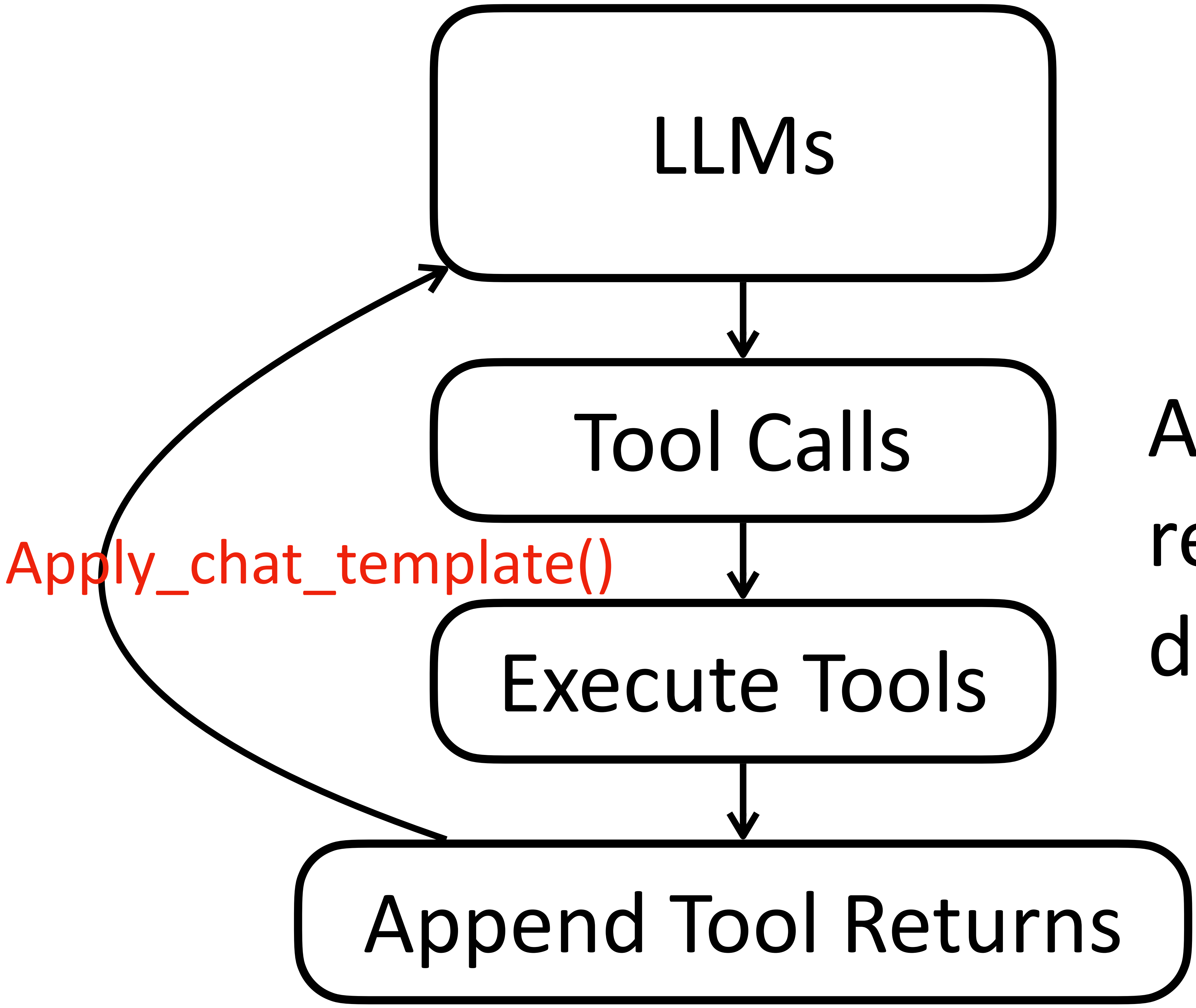
Sure, I’ll check the current weather for you.

[thinking] I need real-time conditions so the user’s route recommendation is accurate. [/thinking]

<tool_call>
{"name": "get_weather", "arguments": {"location": "San Jose, CA, US", "date": "2025-11-07"}}
</tool_call>
<tool_return>
{
  "temperature": 21.5,
  "condition": "clear",
  "humidity": 60,
  "wind_speed": 10,
  "location": "San Jose, CA, US",
  "date": "2025-11-07"
}
</tool_return>

This is the context fed back to the model to continue generation

One-Step Tool Call -> Agents



Agent Loop

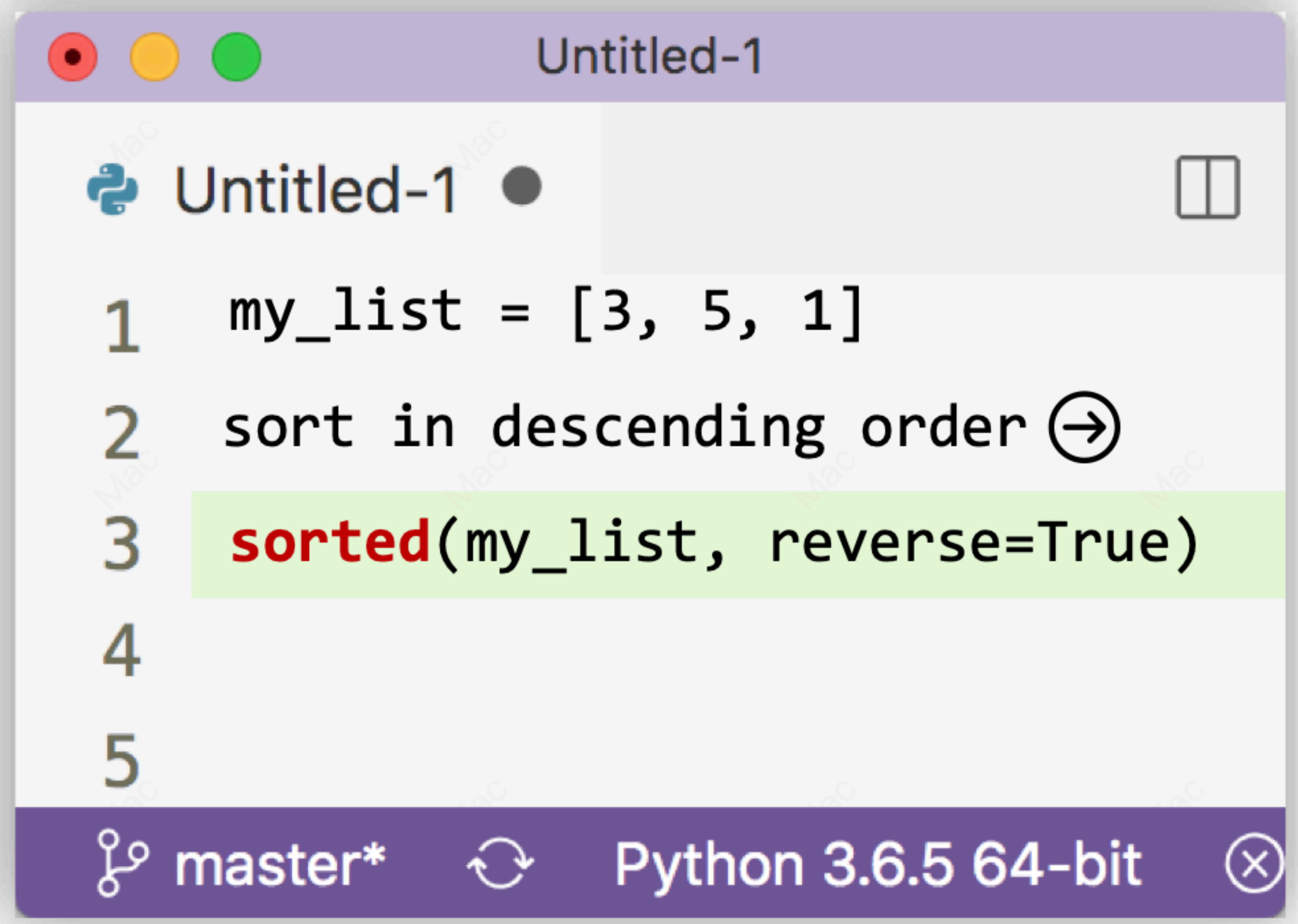
Agent loop ends when the model generates a tex response without tool calling, or sometimes we define a “done” tool for the model to call

One-step Example

```python
def send_messages(messages):
    response = client.chat.completions.create(
        model="deepseek-chat",
        messages=messages,
        tools=tools
    )
    return response.choices[0].message

client = OpenAI(
    api_key="<your api key>",
    base_url="https://api.deepseek.com",
)

tools = [
    {
        "type": "function",
        "function": {
            "name": "get_weather",
            "description": "Get weather of a location, the user should supply a location first.",
            "parameters": {
                "type": "object",
                "properties": {
                    "location": {
                        "type": "string",
                        "description": "The city and state, e.g. San Francisco, CA",
                    }
                },
                "required": ["location"]
            },
        }
    },
]

messages = [{"role": "user", "content": "How's the weather in Hangzhou, Zhejiang?"}]
message = send_messages(messages)
print(f"User>\t {messages[0]['content']}")

tool = message.tool_calls[0]
messages.append(message)

messages.append({"role": "tool", "tool_call_id": tool.id, "content": "24℃"})
message = send_messages(messages)
print(f"Model>\t {message.content}")
```

Why Do We Want Agents

Imagine if things get done by just talking…



Virtual Assistants

- *Set an alarm at 7 AM*
- *Remind me for the meeting at 5pm*
- *Play Jay Chou's latest album*

Natural Language Programming

- *Sort my_list in descending order*
- *Copy my_file to home folder*
- *Dump my_dict as a csv file output.csv*

Training-free Methods for Building Agents

How to Let LLM Become an Agent

Sure, I'll check the current weather for you.

[thinking] I need real-time conditions so the user's route recommendati

<tool_call>
{"name": "get_weather", "arguments": {"location": "San Jose, CA, US", "dat
</tool_call>

The New England Journal of Medicine is a registered trademark of **[QA("Who is the publisher of The New England Journal of Medicine?") → Massachusetts Medical Society]** the MMS.

Out of 1400 participants, 400 (or **[Calculator(400 / 1400) → 0.29]** 29%) passed the test.

The name derives from "la tortuga", the Spanish word for **[MT("tortuga") → turtle]** turtle.

The Brown Act is California's law **[WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.]** that requires legislative bodies, like city councils, to hold their meetings open to the public.

We just need the LLMs to output certain formats of tool calls that we can parse

Just Prompt

Supposing you have the following APIs, answer the given question.

Available APIs:
1. weather(location): for getting the weather information given a location
2. location(): for getting the location of the user.
3. bus_route(start, end): for finding the current bus route from the start position to the end position.
4. count_characters(s): for counting the number of characters in a string s.

Question: Is it okay to go hiking today?

To know the answer, we first need to know the current location of the user.
API call: location() = Seattle.
Next, we need to know the current weather in Seattle.
API call: weather(Seattle) = Cloudy with a chance of rain.

Based on this information, it is not recommended to go hiking today in Seattle.

Available APIs:
1. weather(location): for getting the weather information given a location
2. location(): for getting the location of the user.
3. bus_route(start, end): for finding the current bus route from the start position to the end position.
4. count_characters(s): for counting the number of characters in a string s.

Question: how long does it take to take a bus to Vancouver?

To answer this question, we first need to know the current location of the user.
API call: location() = Seattle.
Next, we need to find the bus route from Seattle to Vancouver.
API call: bus_route(Seattle, Vancouver) = 4 hours.

Therefore, it takes 4 hours to take a bus from Seattle to Vancouver.

We just need the LLMs to output certain formats of tool calls that we can parse

Evaluating Language Agents

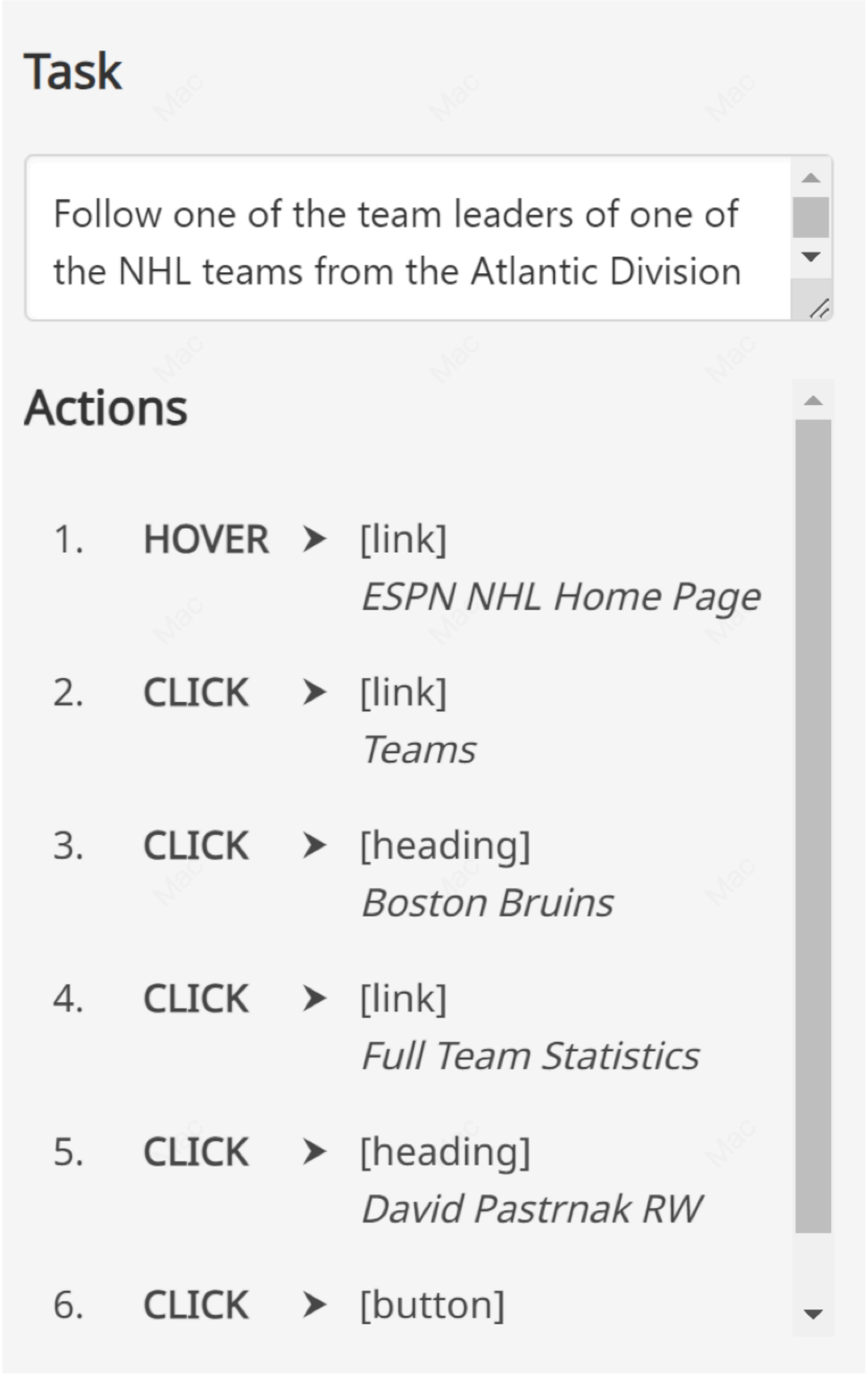
Evaluation of LLM Agents

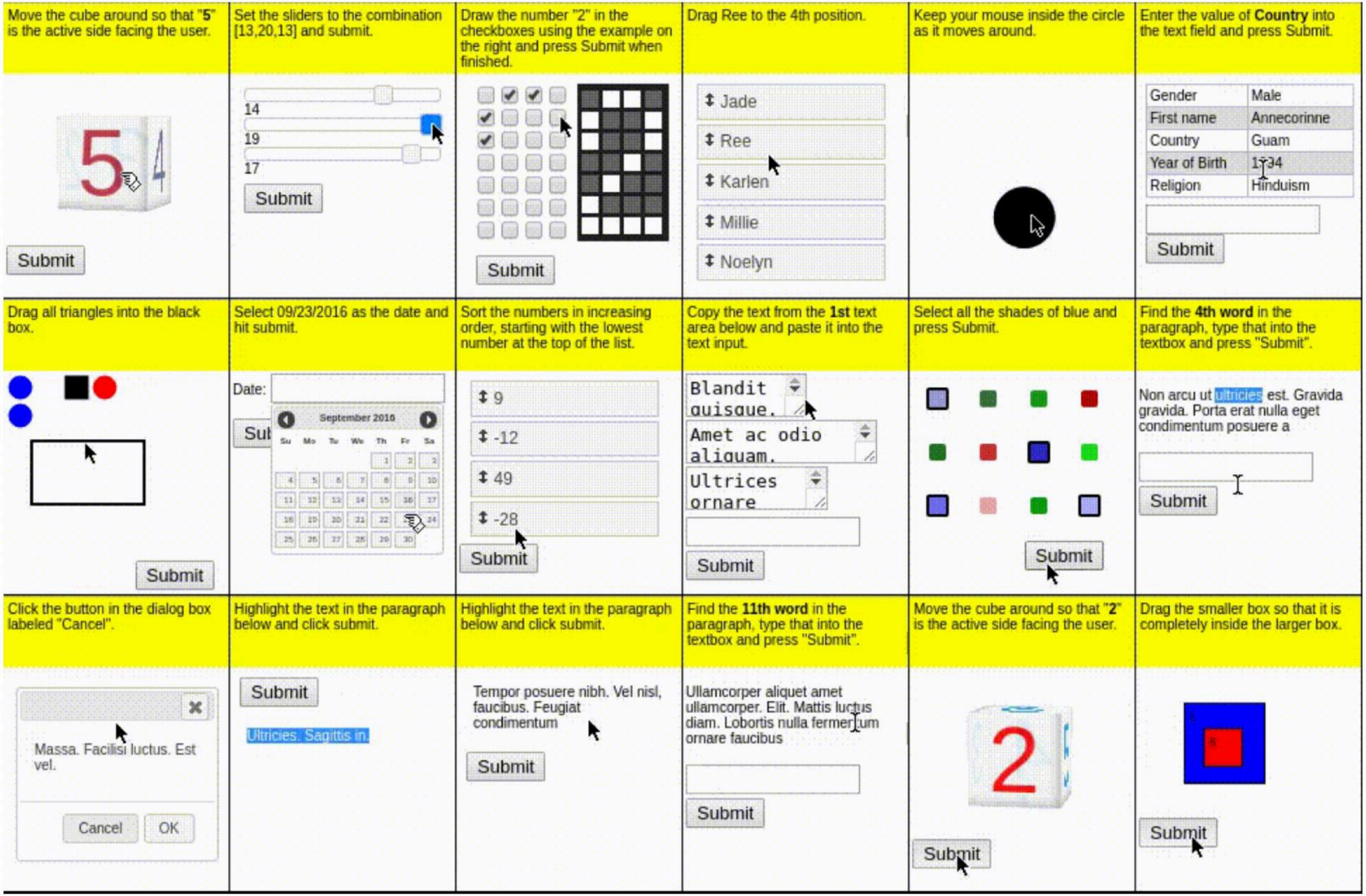
- Simplified environments and basic tasks
- Performance is saturating.

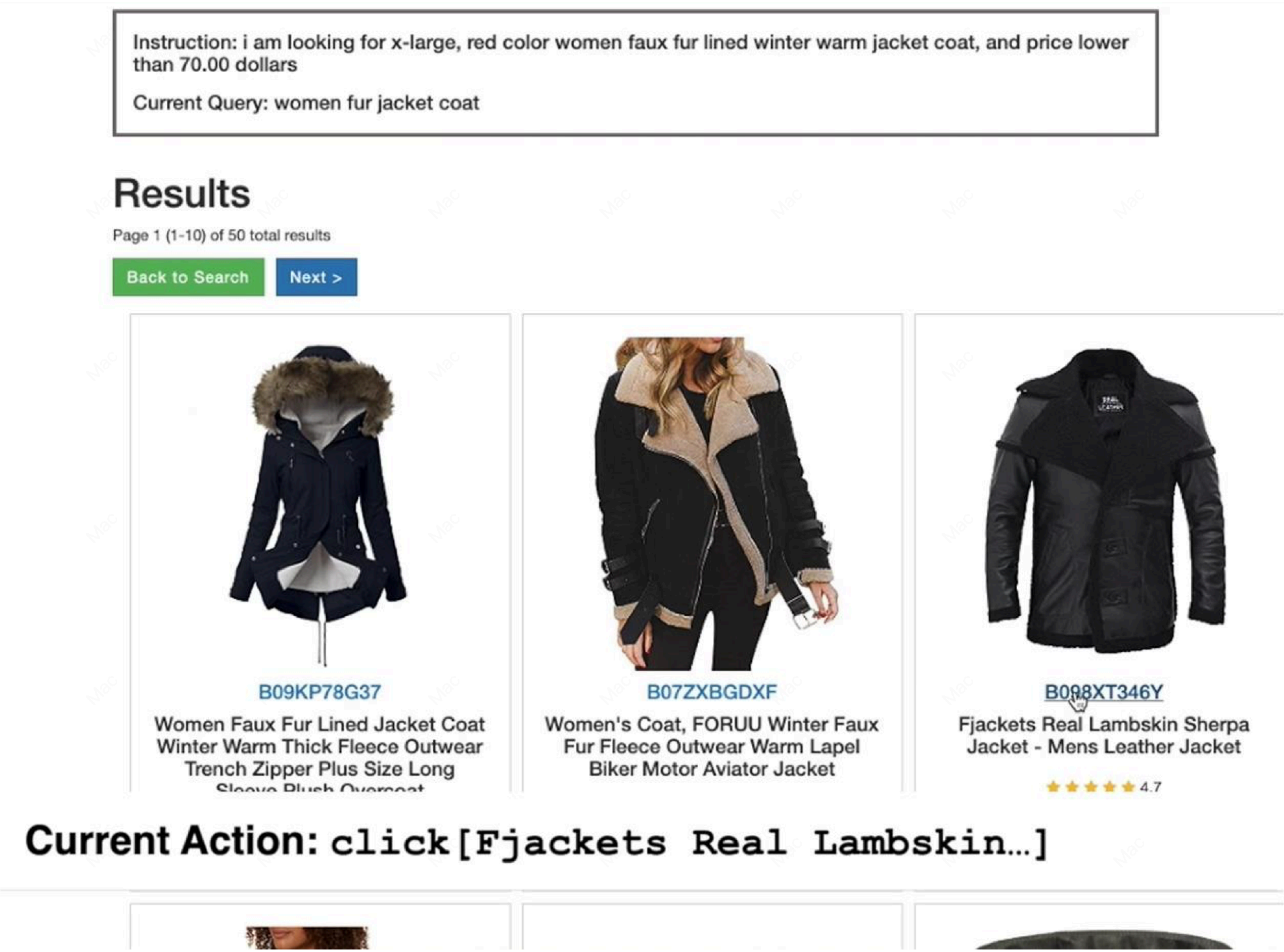
1. Stateless, non interactive environment, e.g. *Mind2Web (Deng et al. 2023)* has only dumped pages.

2. Checking action sequence accuracy (step-wise, surface form only)

3. Simple interactive environment, short horizon, e.g. *WebShop (Yao et al. 2023), MiniWoB++ (Humphreys et al. 2022)*




Move the cube around so that "5" is the active side facing the user.
Submit
Set the sliders to the combination [13,20,13] and submit.
14
19
17
Submit
Draw the number "2" in the checkboxes using the example on the right and press Submit when finished.
Submit
Drag Ree to the 4th position.
Jade
Ree
Karlen
Millie
Noelyn
Keep your mouse inside the circle as it moves around.
Enter the value of Country into the text field and press Submit.
Gender Male
First name Annecorinne
Country Guam
Year of Birth 1934
Religion Hinduism
Submit
Drag all triangles into the black box.
Submit
Select 09/23/2016 as the date and hit submit.
Date:
September 2016
Sort the numbers in increasing order, starting with the lowest number at the top of the list.
9
-12
49
-28
Submit
Copy the text from the 1st text area below and paste it into the text input.
Blandit quisque.
Amet ac odio aliquam.
Ultrices ornare
Submit
Select all the shades of blue and press Submit.
Submit
Find the 4th word in the paragraph, type that into the textbox and press "Submit".
Non arcu ut ultricies est. Gravida gravida. Porta erat nulla eget condimentum posuere a
Submit
Click the button in the dialog box labeled "Cancel".
Massa. Facilisi luctus. Est vel.
Cancel
OK
Highlight the text in the paragraph below and click submit.
Submit
Ultricies. Sagittis in.
Highlight the text in the paragraph below and click submit.
Tempor posuere nibh. Vel nisl, faucibus. Feugiat condimentum
Submit
Find the 11th word in the paragraph, type that into the textbox and press "Submit".
Ullamcorper aliquet amet ullamcorper. Elit. Mattis luctus diam. Lobortis nulla fermentum ornare faucibus
Submit
Move the cube around so that "2" is the active side facing the user.
Submit
Drag the smaller box so that it is completely inside the larger box.
Submit



Instruction: i am looking for x-large, red color women faux fur lined winter warm jacket coat, and price lower than 70.00 dollars
Current Query: women fur jacket coat
Results
Page 1 (1-10) of 50 total results
Back to Search
Next >
B09KP78G37
Women Faux Fur Lined Jacket Coat Winter Warm Thick Fleece Outwear Trench Zipper Plus Size Long
B07ZXBGDXF
Women's Coat, FORUU Winter Faux Fur Fleece Outwear Warm Lapel Biker Motor Aviator Jacket
B098XT346Y
Fjackets Real Lambskin Sherpa Jacket - Mens Leather Jacket
4.7
Current Action: click[Fjackets Real Lambskin...]

Key to Agent Benchmarks

Environment:

- Diverse functionality.
- Rich and realistic content.
- Interactive
- Easily Extendable
- Reproducible

Tasks:

- Long horizon tasks
- Enough difficulty
- Involves multiple websites

Evaluation:

- Reliable metrics
- Encourage final goal rather than partial satisfaction.

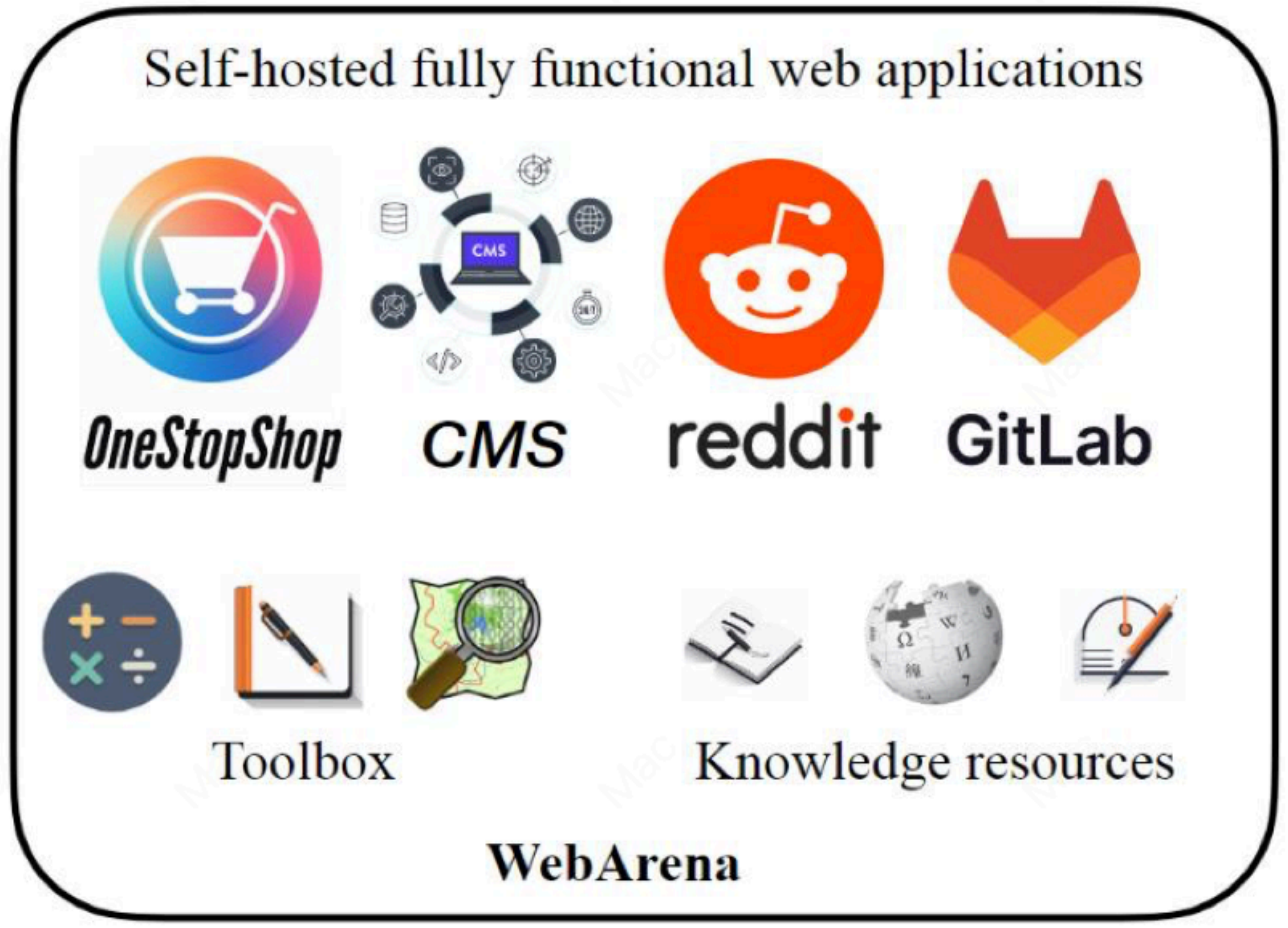
WebArena

Environment:

- Diverse functionality.
- Rich and realistic content.
- Interactive
- Easily Extendable
- Reproducible

A sandbox Internet:

- Open source, production-ready implementation of the websites
- Data populated from real-world websites
- Easily distributable – Dockers, AWS images, etc.

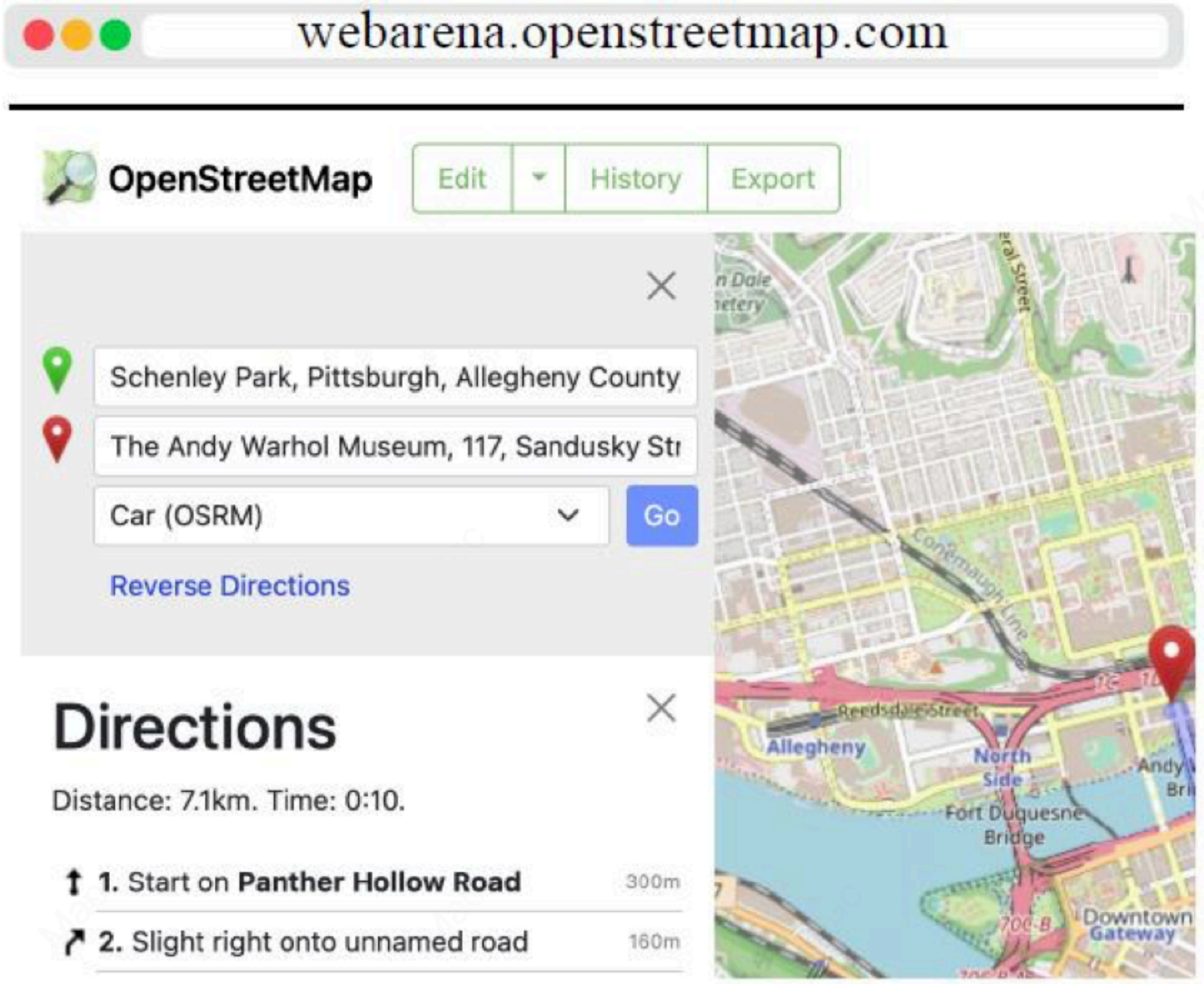


WebArena (Zhou et al. 23')

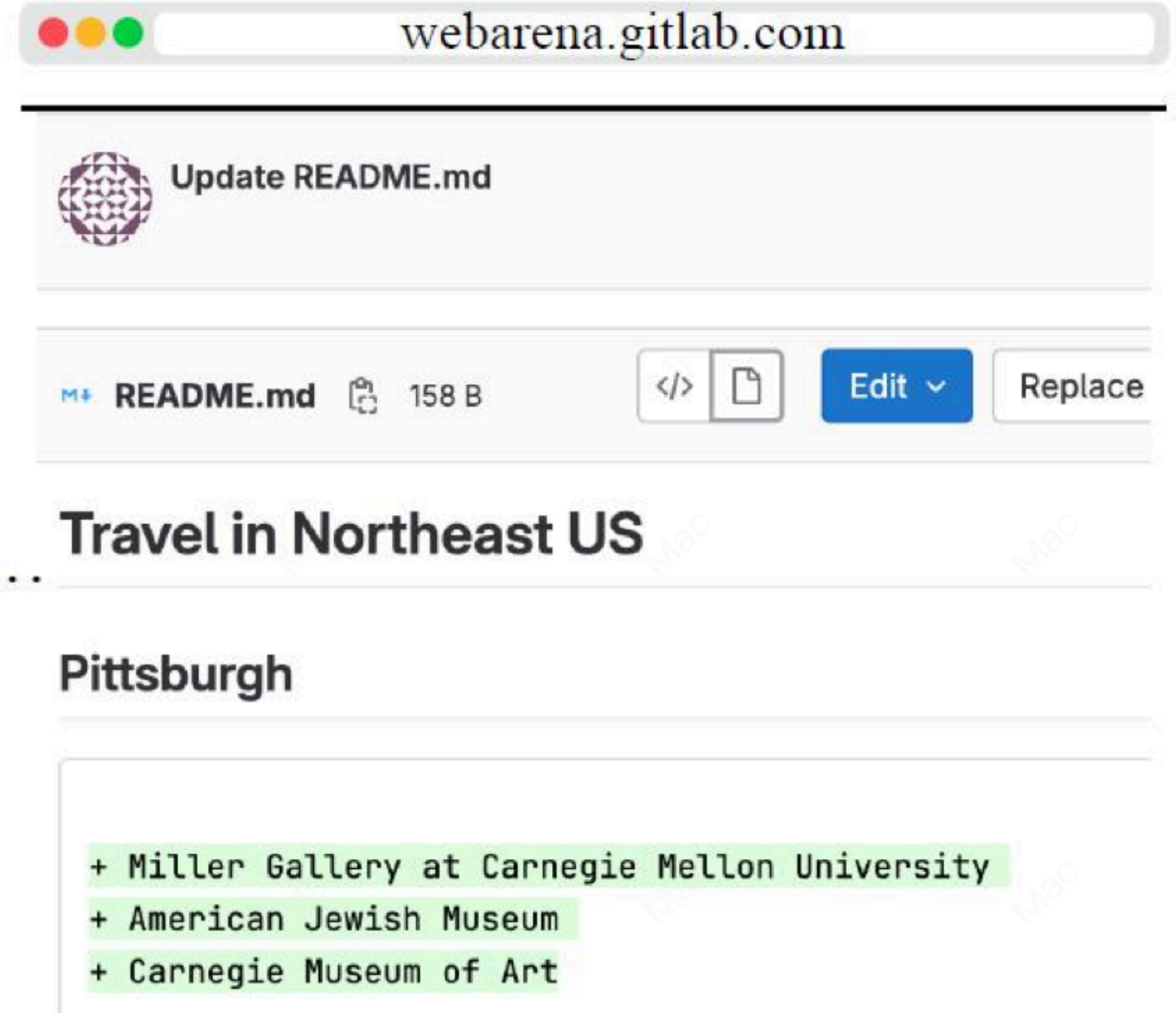
Example Tasks in WebArena

"Create a plan to visit Pittsburgh's art museums with minimal driving distance starting from Schenley Park. Log the order in my "awesome-northeast-us-travel" repository"

Search for museums in Pittsburgh



Search for each art museum on the Map



Record the optimized results to the repo

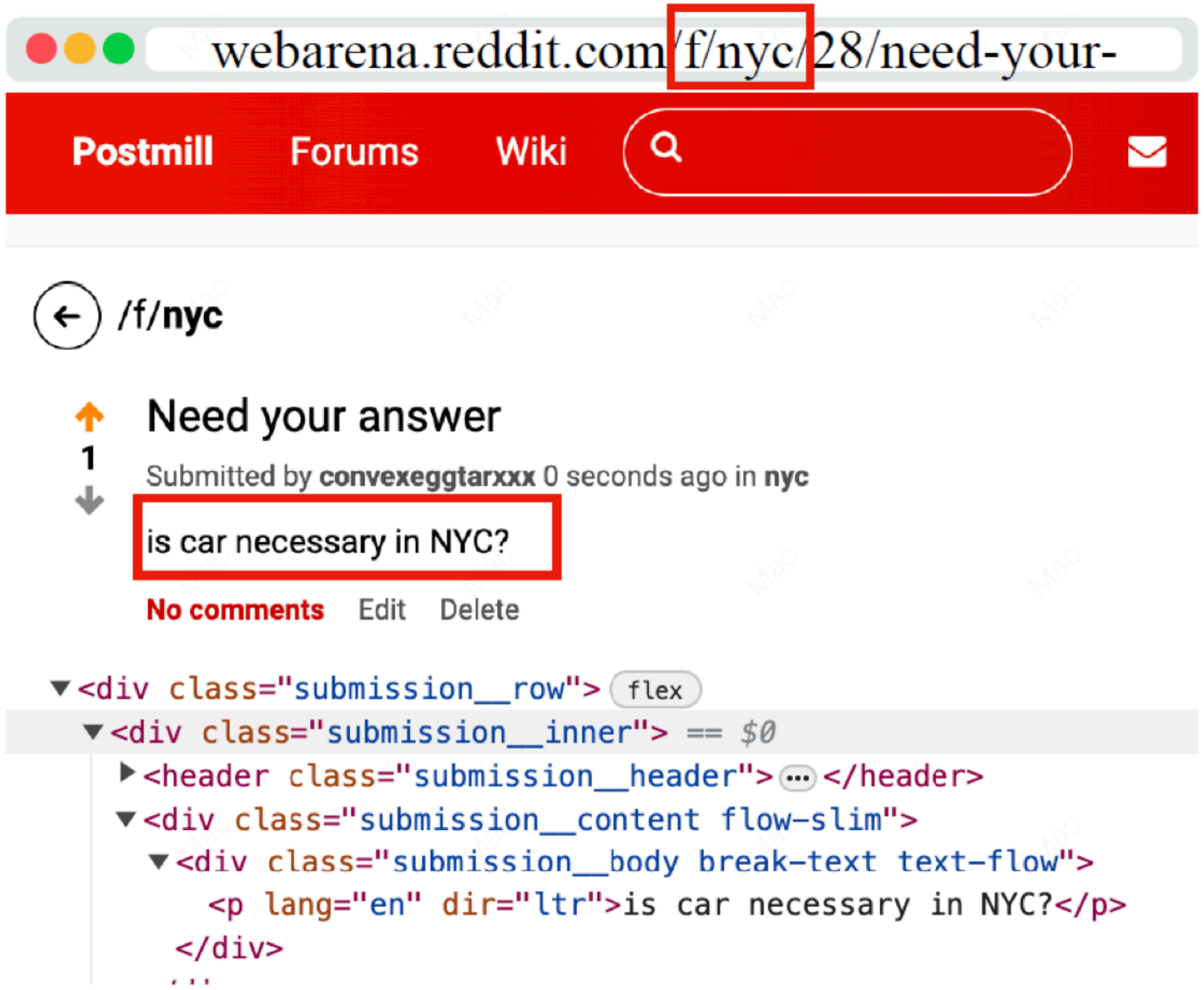
Outcome/Execution-based Evaluation

Goal: directly validate the correctness of the execution

- “When was the last time I bought shampoo?”
- **Directly compare with the annotated answer:** Answer is “Dec 15th, 2022”

Outcome/Execution-based Evaluation

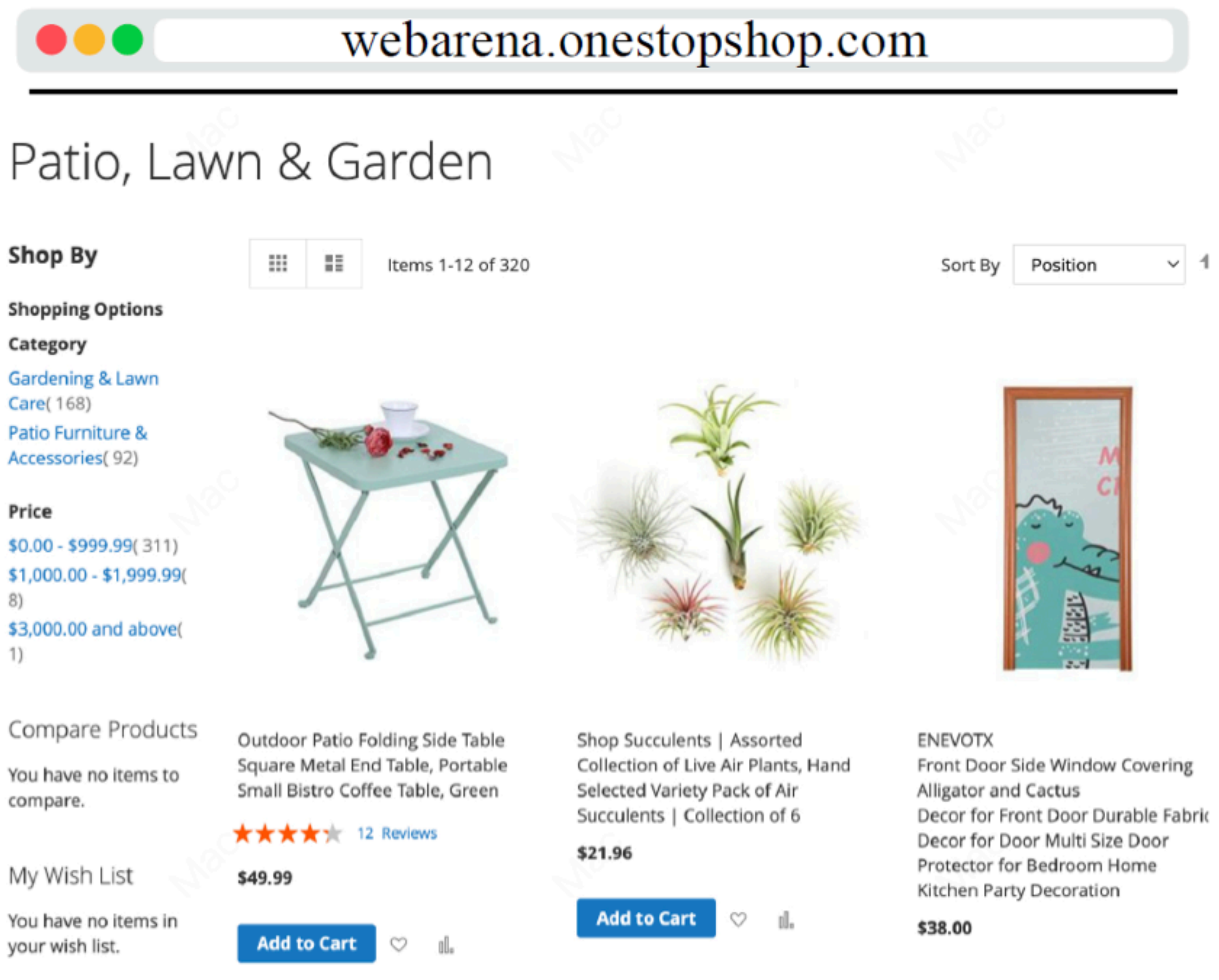
Post my question, “is car necessary in NYC”, in a subreddit where I’m likely to get an answer



“f/nyc” in page.url

“Is car necessary in NYC?” in document.querySelector(“.submission__inner”).outText

Observation & Action Space



Screenshot

webarena.onestopshop.com

```
<li>
 <div>
   <a href="..."><img src="..."></a>
   <div class>
     <a href="...">Outdoor Patio …
</a>
     <div>
        <span>Rating:</span>
        <div>
          <span>82%</span>
        </div>
        <a href="…#reviews">12
<span>Reviews</span></a>
```

Text

webarena.onestopshop.com

```
RootWebArea 'Patio, Lawn ..'
  link 'Image'
    img 'Image'
  link 'Outdoor Patio..'
  LayoutTable ''
     StaticText 'Rating:'
     generic '82%'
     link '12 Reviews'
  StaticText '$49.99'
  button 'Add to Cart' focusable: True
  button 'Wish List' focusable: …
  button 'Compare' focusable: …
```

Accessibility tree

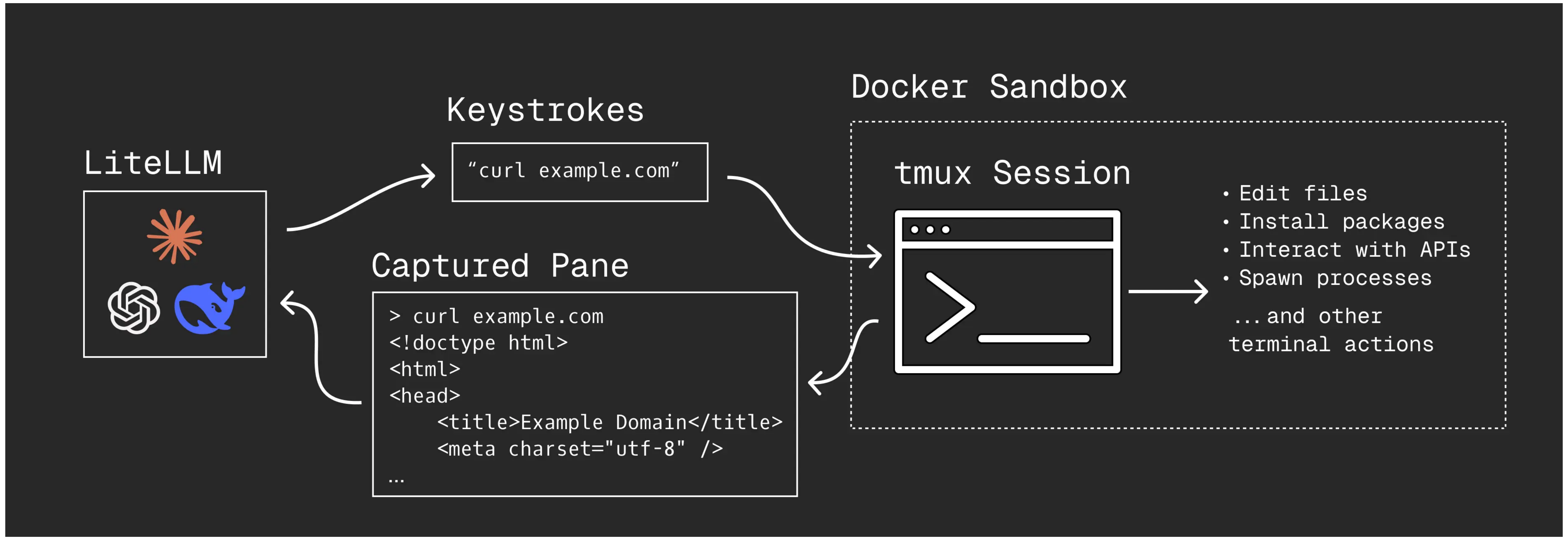
Keyboard: type

Mouse: click, hover, scroll

Browser: New tab, go back

Another type of web agents, GUI agents, directly takes image as input observations

TerminalBench



Terminal-Bench: Possible Agent Actions

- Run shell commands (`ls` , `cd` , `make` , `python` , etc.)
- Manage tmux sessions/panes (`new-session` , `split-window` , `select-pane`)
- Edit files (`vim` , `nano` , `echo > file` , etc.)
- Install/build software (`apt install` , `gcc` , `make`)
- Read & analyze outputs/logs (`cat` , `less` , `grep`)
- Navigate directories and view help (`cd` , `ls` , `--help`)
- Verify or fix results (re-run scripts, check outputs)

Training Methods for Improving Agents

Learning of LLM Agents

- In-Context Learning – Learning from few-shot exemplars
- Supervised Finetuning – Learning From *Experts*
- Reinforcement Learning – Learning from *Environment*

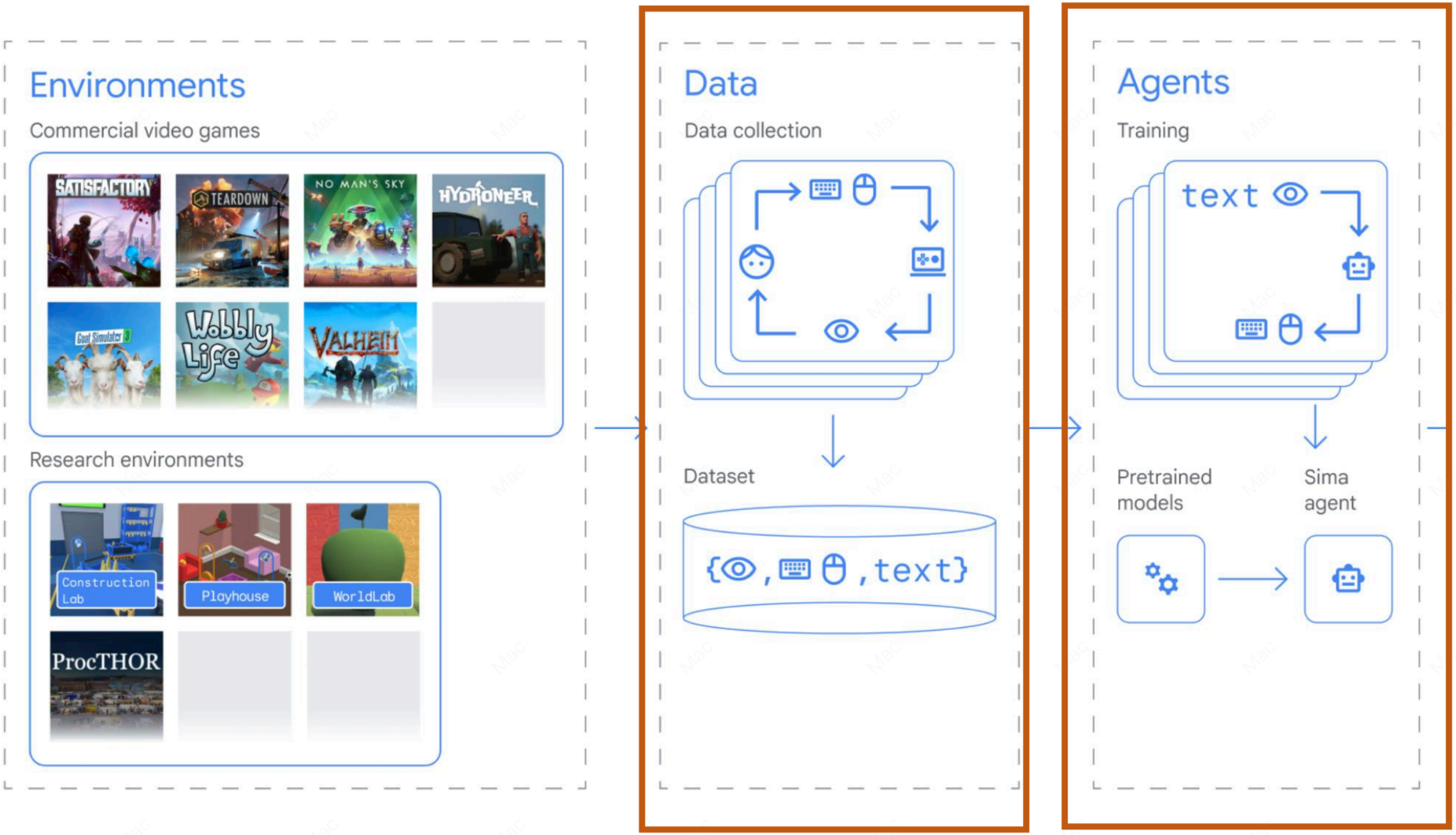
Supervised Finetuning

- Collect large amount of expert trajectories (e.g. from human annotation)

```
task_intent, [(obs_1, action_1), …,(obs_N, action_N)]
```

- Finetune the LLM with standard cross-entropy loss.

Supervised Finetuning



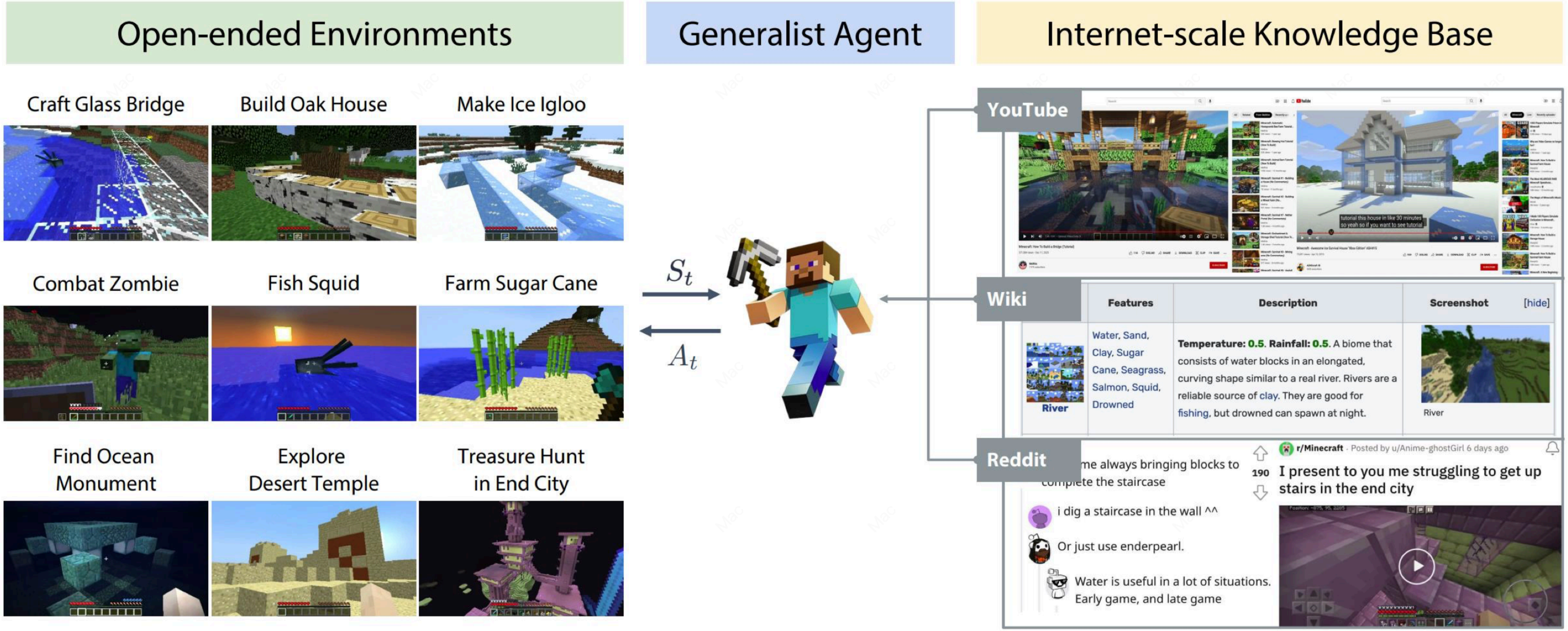
Supervised Finetuning

- Data hungry

- Cannot learn much from failed trajectories
  - `a_1, a_2, a_3, … ,a_10 – Success`
  - `a_1, a_2, a_3, … ,a_10 – Fail (Wasted)`

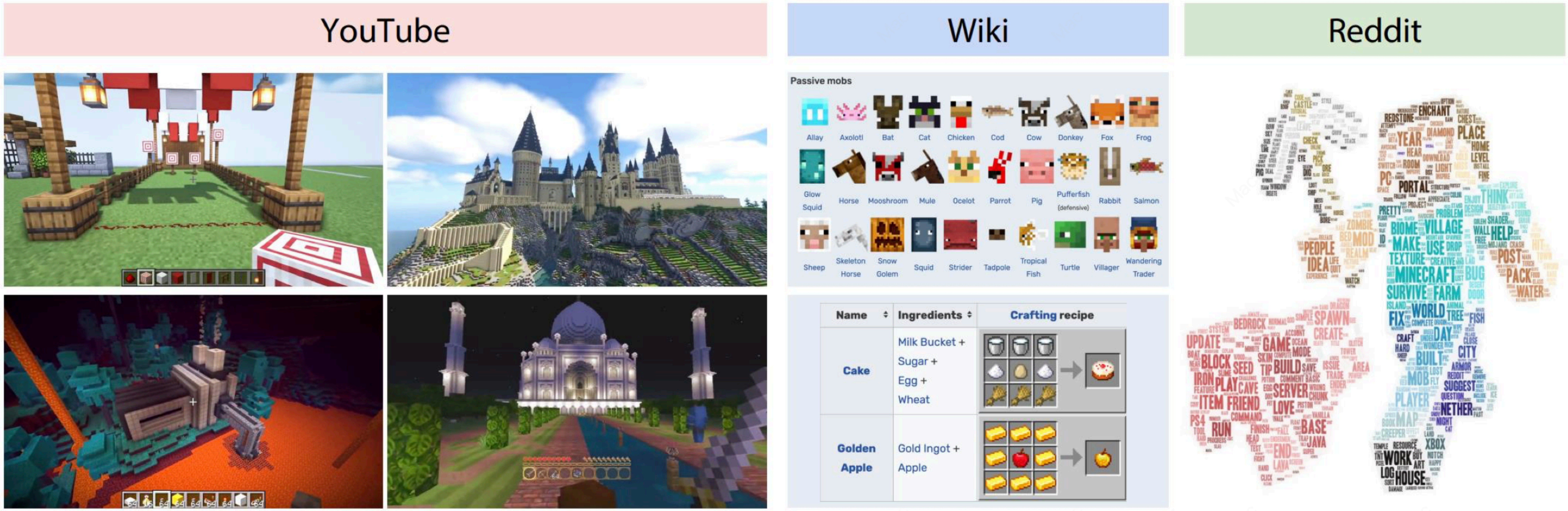
- Need human trajectory?
  - Data augmentation techniques

Create More Training Data



Data Augmentation

- Continue pre-train on large amount of data automatically mined
- Even noisy, not clear trajectories, provide domain adaptation.

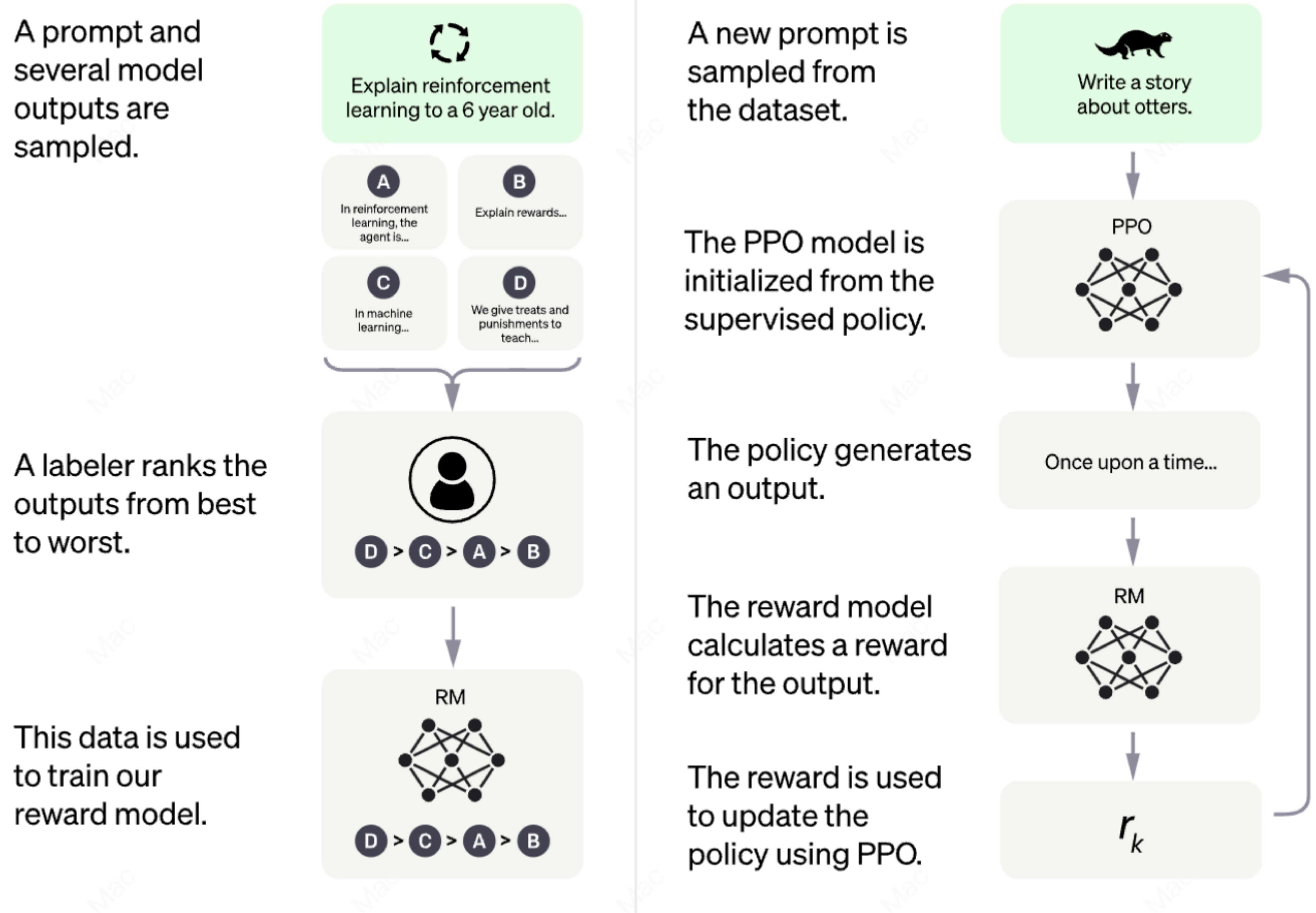


MineDojo, Fan et al. 22’
Don’t Stop Pretraining, Gururangan et al., 20’

Reinforcement Learning

Lots of on-going research in this area!

Recall RLHF: Reinforcement Learning from Human Feedback:



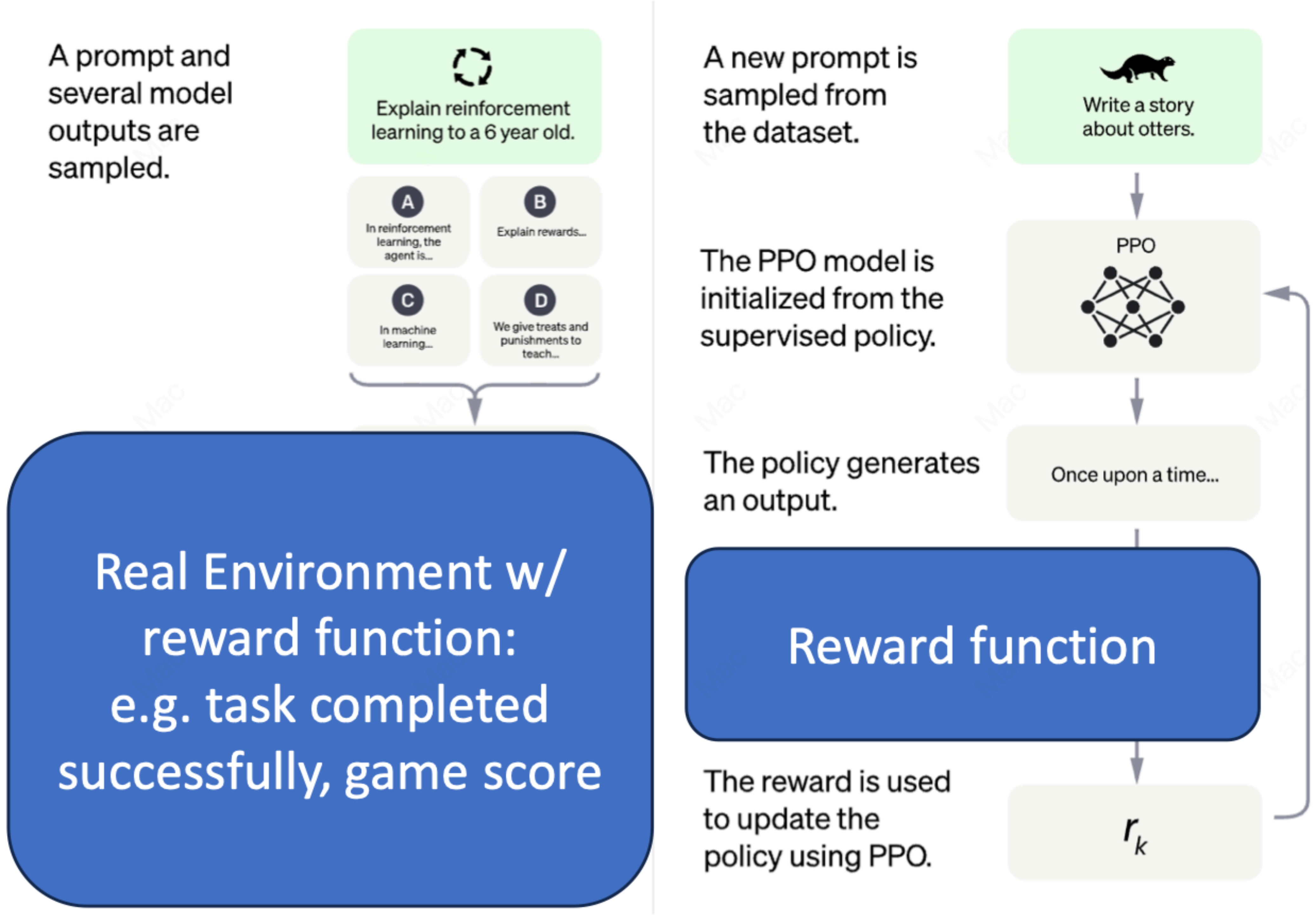
RLHF, Ouyang, et al. 22’

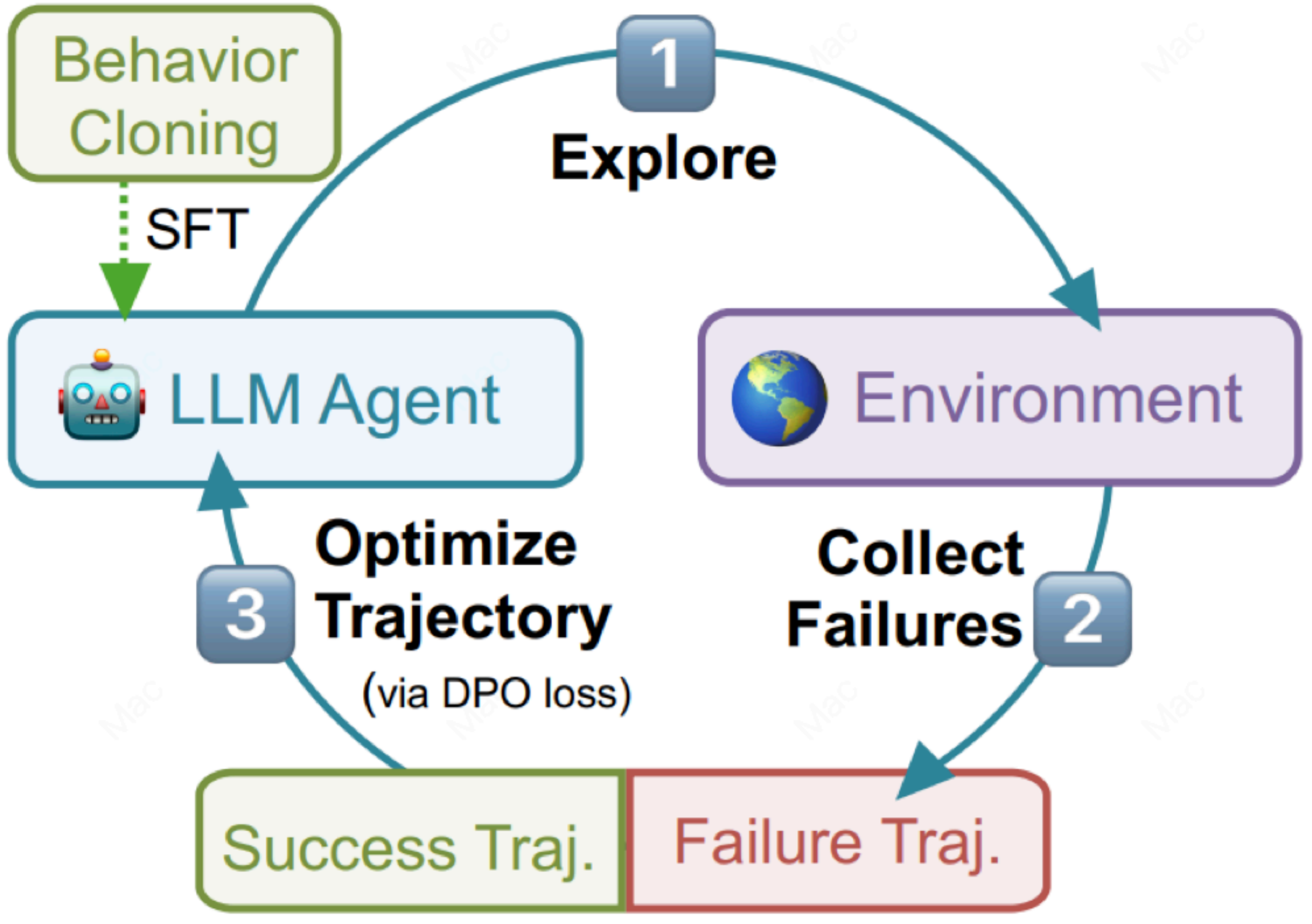
Reinforcement Learning

Compared to RLHF:

Given environment, *reward function*

(trajectory, reward) pairs without human





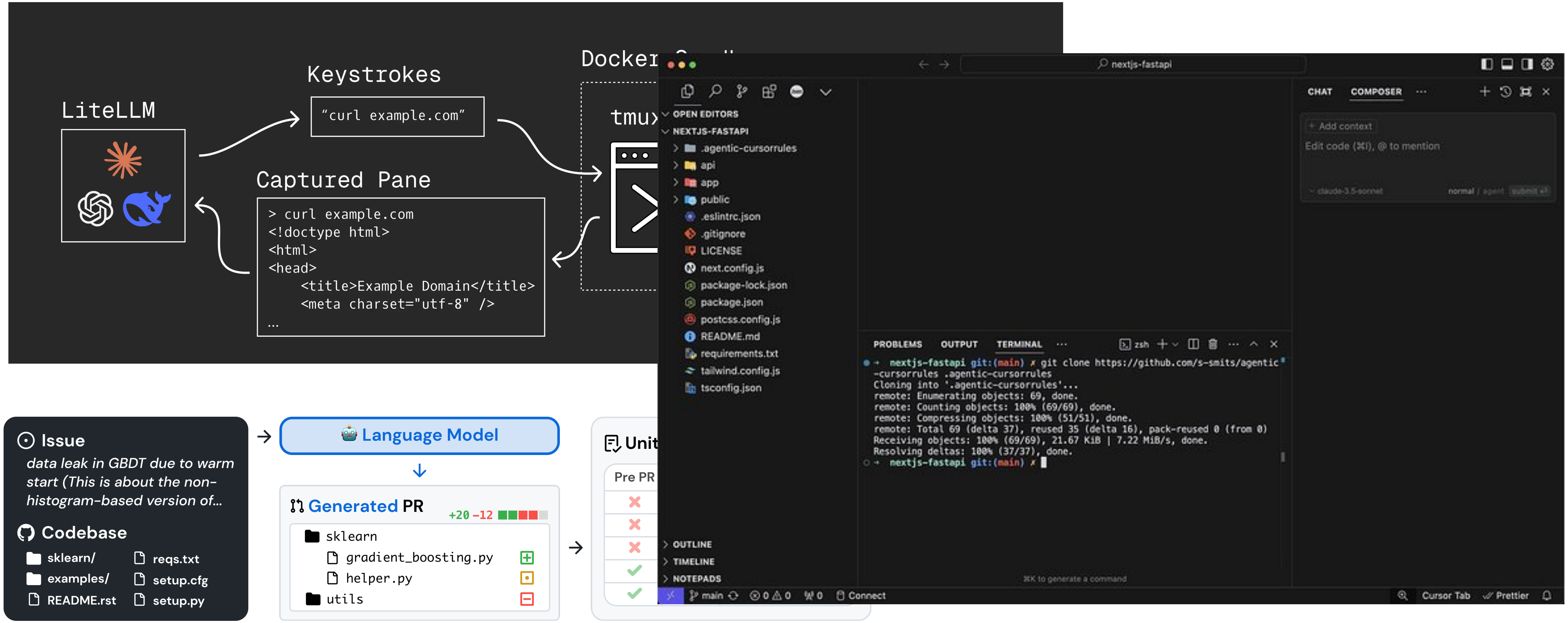
Trial and Error (Song et al. 24')

Reinforcement Learning

- Closed loop, interactive environment
- Need good reward functions
  - What if the task success/fail is not easy to automatically assess?
- Need good initial models
  - Has decent basic knowledge ability, sparse rewards
- Scalability
  - The environment takes 10 seconds to env.step()
  - The reward function takes 100 seconds to get a scalar reward

RL Environments

Environments and benchmarks typically come together



Research and Products are really close nowadays, and we can directly RL in real, product-level environments

Thank You!