

Web Agents

October 30th

Project goal recap

- An agent on top of a LLM or VLM can solve tasks by navigating the web.
 - But simple agents like this will struggle with complex or long horizon tasks
- RL might make agents better
 - But it's inefficient and relies on a reasonable initial performance to improve from
- Supervised fine-tuning has the potential to significantly improve web agent performance on challenging tasks
 - But where is the data?

Solution: watch people complete tasks.



Browser Agent Benchmarks

- WebVoyager (real world websites)
- Web Bench
- BrowseComp
- Mind2Web
- WEBARENA
- GAIA
- WebDS

Enterprise Benchmarks

- Work Bench(real world websites)
- Spreadsheet Bench
- DocBench
- MailBench
- SlideBench
- CalendarBench

Potential opening

- There was no cross App interaction across the Gsuite , M365, slack.

Project Update

- We're planning focus to Work Arena as our main environment
- And to try this out before trying any of the cross App
- Using our custom Chrome extension to record real human trajectories
 - HTML snapshots
 - Accessibility tree
 - Video timestamps
 - Action logs
- Goal: Build a high-quality dataset to fine-tune and ev



WorkArena Tasks

Task Levels and Their Characteristics

- L1 Tasks (WorkArena):
 - ~200 tasks
 - Single-step, simple actions (e.g., filling a form, clicking a button, simple queries)
 - Designed to test basic perception, action grounding, navigation.
- L2 Tasks (WorkArena++):
 - ~300 tasks
 - Multi-step procedures with intermediate planning (e.g., navigating lists, combining filter and retrieval, submitting forms in sequence)
 - Tests for short-horizon reasoning, data extraction, and manipulation.
- L3 Tasks (WorkArena++):
 - ~182 tasks
 - Require complex reasoning, compositionality (e.g., planning multi-step workflows, handling conditional logic, integrating information from multiple locations within ServiceNow)
 - Test long-horizon planning & generalization.

Task Diversity

Task Types Include:

- Catalog ordering (e.g., order an iPad Pro, submit an IT service request)
- Filtering and sorting lists (users, incidents, hardware)
- Dashboards (retrieving and combining information from multiple visualizations)
- Multi-field form filling
- Approval routing and basic workflow management
- Information retrieval, search, and report submission

Evaluation Protocol

Automatic validation:

- Each task has a reference solution and validation logic.

Seed diversity:

- For each task, they will have multiple seed variations.

BrowserGym Leaderboard

WorkArena-L1	WebArena	WorkArena-L1	WorkArena-L2	WorkArena-L3	MiniWoB	WebLINX	VisualWebArena	AssistantBench
GenericAgent-GPT-5	-	79.10	69.40	11.50	71.50	-	-	-
GenericAgent-Claude-4-Sonnet	-	63.30	40.40	-	70.70	-	-	-
GenericAgent-GPT-5-mini	-	60.60	47.70	-	71.00	-	-	-
GenericAgent-GPT-o1-mini	28.60	56.70	14.90	0.00	67.80	12.50	-	6.90
GenericAgent-Claude-3.5-Sonnet	36.20	56.40	39.10	0.40	69.80	13.70	21.00	5.20
GenericAgent-GPT-oss-120b	-	50.90	11.50	-	66.40	-	-	-
GenericAgent-o3-mini	-	48.20	-	-	-	-	-	-
GenericAgent-GPT-4o	31.40	45.50	8.50	0.00	63.80	12.50	26.70	4.80
GenericAgent-Llama-3.1-405b	24.00	43.30	7.20	0.00	64.60	7.90	-	3.90
GenericAgent-GPT-5-nano	-	40.60	3.40	-	64.80	-	-	-
GenericAgent-GPT-oss-20b	-	38.50	2.60	-	64.00	-	-	-
GenericAgent-AgentTrek-1.0-32b	22.40	38.29	2.98	0.00	60.00	-	-	-
GenericAgent-Llama-3.1-70b	18.40	27.90	2.10	0.00	57.60	8.90	-	2.80
GenericAgent-GPT-4o-mini	17.40	27.00	1.30	0.00	56.60	11.60	16.90	2.10

Baseline Experiment

Agents	WorkAreana Tasks		
	L1	L2	L3
GenericAgent GPT5	79.1	69.4	11.5
Generic Agent gpt-4o-mini	27	1.3	0
Generic Agent gpt-4o-mini	31.8	1.8	0

Methodology

- Collect → Human demonstrations using the browser extension
- Post-process → (State, Action, Next State) triplets from HTML, AxTree, and video.
- Train → Fine-tune the base LLM with supervised learning.
- Evaluate → Benchmark on WorkArena
- Analyze → Check generalization and overfitting effects.

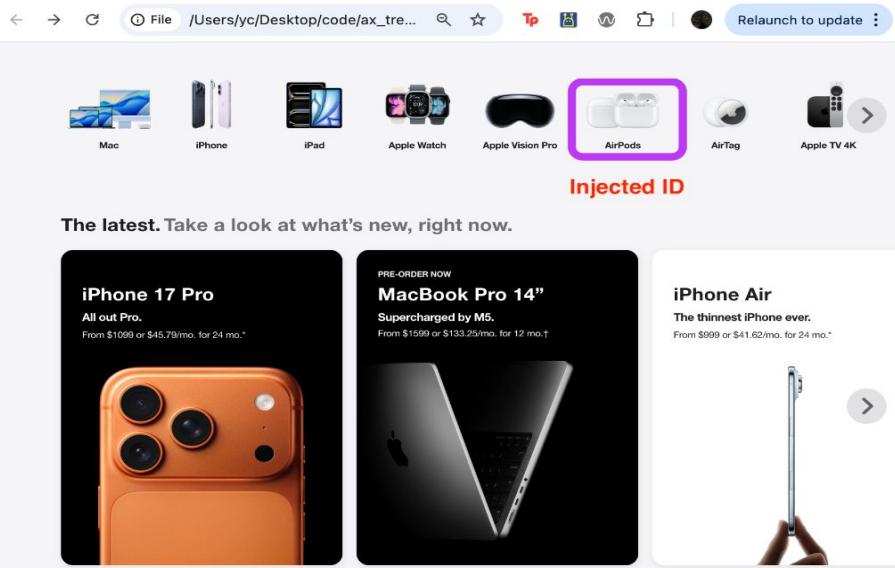
Why This Is Novel

Turning the stream of data into meaningful training samples

- Most prior work uses synthetic or scripted web trajectories.
- We collect natural, human trajectories directly from task executions.
- Few papers explore fine-tuning on domain-specific enterprise web data.



Debugging BrowserGym



The latest. Take a look at what's new, right now.

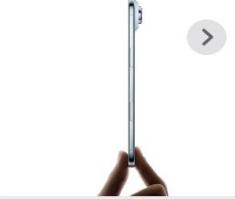
iPhone 17 Pro
All out Pro.
From \$1099 or \$45.79/mo. for 24 mo.*



PRE-ORDER NOW
MacBook Pro 14"
Supercharged by M5.
From \$1599 or \$133.25/mo. for 12 mo.†



iPhone Air
The thinnest iPhone ever.
From \$999 or \$41.62/mo. for 24 mo.*



Help is here. Whenever and however you need it.

APPLE SPECIALIST

Shop one on one with a Specialist. Online or in a store.



Shop with a Specialist over video.

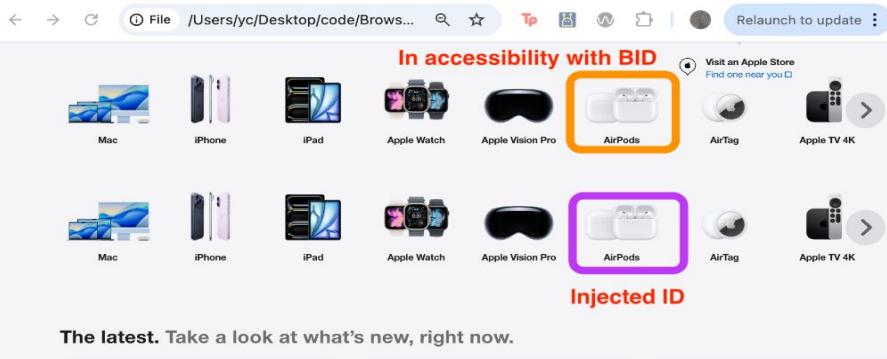
Choose your next device in a guided, one-way video session.



TODAY AT APPLE

Explore App Intelligence

Come try it for yourself Apple Store.



In accessibility with BID

Visit an Apple Store
Find one near you □

The latest. Take a look at what's new, right now.

iPhone 17 Pro
All out Pro.
From \$1099 or \$45.79/mo. for 24 mo.*



PRE-ORDER NOW
MacBook Pro 14"
Supercharged by M5.
From \$1599 or \$133.25 /mo. for 12 mo. †



iPhone Air
The thinnest iPhone ever.
From \$999 or \$41.62 /mo. for 24 mo. *



The latest. Take a look at what's new, right now.

iPhone 17 Pro

All out Pro.

From \$1099 or \$45.79/mo. for 24 mo.*



PRE-ORDER NOW
MacBook Pro 14"

Supercharged by M5.

From \$1599 or \$133.25/mo. for 12 mo. †



iPhone Air

The thinnest iPhone ever.

From \$999 or \$41.62 /mo. for 24 mo. *



Amazon.com only undergo minor changes

A screenshot of the Amazon search results for "chocolate". The search bar at the top shows "chocolate". The main content area displays a grid of products under the heading "Everyday is sweeter with Hershey's". Products shown include Hershey's Nuggets Assorted Chocolate, Hershey's, Kit Kat and Reese's Assorted, and Hershey's and Reese's Miniatures Assorted. On the left sidebar, there are sections for "Popular Shopping Ideas" (Vegan, Sugar Free, Sea Salt, Organic), "Eligible for Free Shipping" (Free Shipping by Amazon, Get FREE Shipping on eligible orders shipped by Amazon), "Delivery Day" (Get It Today, Get It by Tomorrow), "Price" (\$0 - \$1,450+), and "Deals & Discounts" (All Discounts, Today's Deals). A "Sort by: Featured" dropdown is also present.

A screenshot of the same Amazon search results for "chocolate", likely taken later. The interface has undergone minor visual changes. The "Popular Shopping Ideas" section now includes "Organic" and a "See more" link. The "Eligible for Free Shipping" section has been renamed "Delivery Day". The "Price" filter has been updated to show price ranges from Up to \$10 to \$30 & above. The "Deals & Discounts" section now includes "Today's Deals". The overall layout and product grid remain similar to the first screenshot.

A screenshot of the Amazon search results for "chocolate" showing significant visual changes. The "Popular Shopping Ideas" section is now titled "Popular Shopping Ideas" and includes "Vegan", "Sugar Free", "Sea Salt", "Organic", and a "See more" link. The "Eligible for Free Shipping" section is now titled "Delivery Day" and includes "Get It Today" and "Get It by Tomorrow". The "Price" filter is now a range from \$0 - \$1,450+. The "Deals & Discounts" section is now titled "Deals & Discounts" and includes "All Discounts" and "Today's Deals". The "From Our Brands" section lists "Amazon Brands", "SNAP EBT", and "Customer Reviews". The "Customer Reviews" section includes a "★★★★★ & Up" link. The "All Top Brands" section lists "Top Brands", "Brands", and "All Top Brands". The "Brands" section includes "Lindt", "HERSHEY'S", "Cadbury", "Dove", "M&M's", "Reese's", and "VALRHONA". The "Type" section includes "Milk", "Dark", and "White". A purple box highlights the "SNAP EBT" link in the "Customer Reviews" section. Another purple box highlights the "Add to cart" button in the "Customer Reviews" section. The overall layout is more modern and organized compared to the previous screenshots.

How it look like on our target website

The screenshot shows a ServiceNow Admin Page for a 'Loaner Laptop' request. The page has a header with 'MIGHTY CAPITAL' and a search bar. Below the header, there's a breadcrumb trail: 'Service Catalog > Hardware > Loaner Laptop'. A message states: 'Short term, while computer is repaired/imaged. Waiting for computer order, special projects, etc. Training, special events, check-in process'. A note says: 'Did you break your laptop? Maybe lost it? Need a temporary loaner? We can help.' Another note specifies: 'In order to take advantage of a loaner notebook computer, you must meet company eligibility requirements per the Notebook Loaner Policy'. A note about availability: 'Loaner laptops will be provided based on what devices are available.' On the right, there's a sidebar with an 'Order this Item' section showing 'Quantity' (1) and 'Delivery time' (2 Days), with 'Order Now' and 'Add to Cart' buttons. Below that is a 'Shopping Cart' section stating 'Empty'. There are also input fields for 'When do you need it?' and 'How long do you need it for?', both currently set to '1 day'.

Live ServiceNow Admin Page



Target elements are in iFrame

The image displays two separate browser windows side-by-side, both showing the same service catalog page for a 'Loaner Laptop'.

Left Browser Window:

- Title Bar:** empmassimo23.service-now.com/...
- Header:** MIGHTY CAPITAL All Loaner Laptop
- Breadcrumbs:** Service Catalog > Hardware > Loaner Laptop
- Text:** Short term, while computer is repaired/imaged. Waiting for computer order, special projects, etc. Training, special events, check-in process
- Text:** Did you break your laptop? Maybe lost it? Need a temporary loaner? We can help.
- Text:** In order to take advantage of a loaner notebook computer, you must meet company eligibility requirements per the Notebook Loaner Policy
- Text:** Loaner laptops will be provided based on what devices are available.
- Form Fields:** When do you need it? (input field), How long do you need it for? (dropdown menu showing "1 day"). Both fields are highlighted with a red box.
- Order Item Section:** Order this Item, Quantity (1), Delivery time (2 Days), Order Now, Add to Cart.
- Shopping Cart:** Empty

Right Browser Window:

- Title Bar:** /Users/yc/Desktop/code/Brows...
- Header:** Service Catalog Hardware Loaner Laptop
- Breadcrumbs:** Back
- Text:** Short term, while computer is repaired/imaged. Waiting for computer order, special projects, etc. Training, special events, check-in process
- Text:** Did you break your laptop? Maybe lost it? Need a temporary loaner? We can help.
- Text:** In order to take advantage of a loaner notebook computer, you must meet company eligibility requirements per the Notebook Loaner Policy
- Text:** Loaner laptops will be provided based on what devices are available.
- Form Fields:** When do you need it? (input field), How long do you need it for? (dropdown menu showing "1 day"). Both fields are highlighted with a red box.
- Order Item Section:** Order this Item, Quantity (1), Delivery time (2 Days), Order Now, Add to Cart.
- Shopping Cart:** Empty

Prompt to generate similar task data

I will provide you original screenshot with the interacted element highlighted in a red box, raw **HTML**, **original task**, and **correct response**. Your job is to generate a response based on the same screenshot and HTML, but with a **different task**.

You need to generate a JSON object inside "```" "```". The object has two key "element" and "action".

1. Element: the element's nodeID
2. Action: an action within action space defined below.

Action Space:

- `click`: This action clicks on an element with a specific id on the webpage.
- `type [content] [press_enter_after=0|1]`: Use this to type the content into the field. By default, the "Enter" key is pressed after typing unless press_enter_after is set to 0.
- `hover`: Hover over an element.
- `press [key_comb]`: Simulates the pressing of a key combination on the keyboard (e.g., Ctrl+v).
- `scroll [down/up]`: Scroll the page up or down. You need to output the command like scroll [down] to scroll down.
- `goto [url]`: Navigate to a specific URL.
- `go_back`: Navigate to the previously viewed page.
- `go_forward`: Navigate to the next page (if a previous 'go_back' action was performed).
- `stop [answer]`: Issue this action when you believe the task is complete. If the objective is to find a text-based answer, provide the answer in the bracket. If you believe the task is impossible to complete, provide the answer as "N/A" in the bracket.

Example response "```"

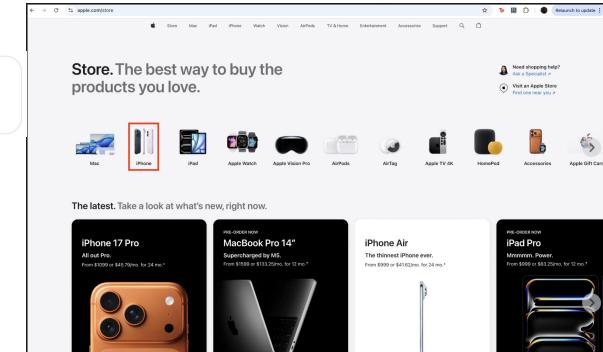
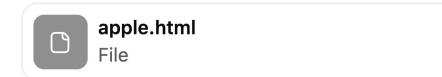
```
{  
  "element" : "document.querySelector(\"body > 123\").querySelector(\"#item\").querySelector(\"button\")"  
  "action" : "click"  
}  
```
```

The original task is "buy an iPhone".

The response is "```"

```
{
 "element" : "document.querySelector(\"body > 123\").querySelector(\"#item\").querySelector(\"button\")" ,
 "action" : "click"
}
```
```

Now, generate a response with new task "**Buy an iPad**".



Verification

Why do we need to verify?

Verification ensures that both **data and actions** produced by the agent are **accurate, interpretable, and correctable** - across *training* and *deployment*.

Garbage in <=> Garbage out

When do we verify?

During Data Collection / Training:

1. Validate event correctness
2. Filter “non consequential events”
3. Grounded learning/ Action - Consequence - (state, action, next_state)

During Inference / Runtime

1. Real-time action validation - 200 OK
2. Self-correction and recovery
3. State tracking and awareness
4. Reference Replay Comparison 🤝

How do we get started on Verification?

Step Verification:

Validation of each **atomic browser action** against an **immediate, observable ground truth**.

Flow-Level Verification:

Evaluation of the entire **multi-step trajectory** against a clearly defined **task goal and reference outcome**

Do we need both? - Ideally yes

Step-level = **local accuracy** → catches micro-failures early

Flow-level = **global correctness** → ensures end-to-end success

But what exactly is getting verified?

1. Action preconditions

If element is present in the DOM, visible, interactable (`element.clickable == true`).

2. Action intent match

The selected element's semantic role, label, or text aligns with the expected action description (from the plan or prompt).

3. Action execution result

measurable post-state change (e.g., DOM mutation, network request, navigation event, or updated field value).

But what are we verifying against?

Source of truth

1. Browser runtime observations (DOM state, URL, network logs).
2. Deterministic expected conditions in the prompt/ terminal states
3. Reference Replay

In Summary

Signal Type	What It Measures	Verification Purpose
Selector validity	Existence, Visibility, Interactability	Detects broken or outdated selectors
Event return code	Whether browser API (e.g., <code>click()</code>) returned <code>OK</code>	Confirms the action executed without exceptions
HTTP status	Status of triggered network requests (e.g., 200 OK)	Validates that backend accepted the operation
DOM diff	Change in DOM tree before vs after the action	Confirms a meaningful state change occurred
Screenshot hash	Pixel-level checksum difference	Detects visible page updates or failures

Discussion

- Help us design some experiments
 - How to verify that the extension is collecting what we want it to and aligns with BrowserGym observations
 - How to determine good inputs for the underlying model(s) without overwhelming ourselves with a billion different possibilities



Spot the difference?