

VisualWebArena: Evaluating Multimodal Agents on Realistic Visually Grounded Web Tasks

Jing Yu Koh Robert Lo* Lawrence Jang* Vikram Duvvur*
Ming Chong Lim* Po-Yu Huang* Graham Neubig Shuyan Zhou
Ruslan Salakhutdinov Daniel Fried
Carnegie Mellon University
{jingyuk, rsalakhu, dfried}@cs.cmu.edu



ACL 2024

Bangkok, Thailand

WEBARENA: A REALISTIC WEB ENVIRONMENT FOR BUILDING AUTONOMOUS AGENTS

Shuyan Zhou* Frank F. Xu*
Hao Zhu† Xuhui Zhou† Robert Lo† Abishek Sridhar† Xianyi Cheng
Tianyue Ou Yonatan Bisk Daniel Fried Uri Alon Graham Neubig

Carnegie Mellon University
{shuyanzh, fangzhex, gneubig}@cs.cmu.edu



ICLR
2024

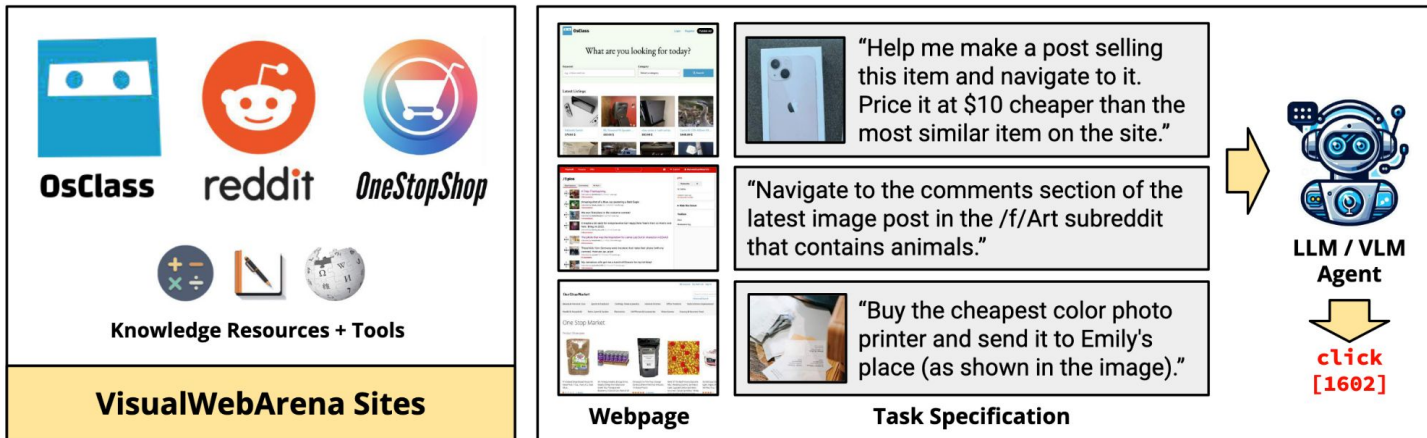
Presenter: Min Shi, Chieh-Yun Chen

Outline

1. One page summary the paper
2. What is autonomous agent ?
3. How to evaluate an Autonomous Agent ?
4. How to formulate an agent ?
5. Experiments
6. Conclusion
 - Key findings
 - Limitations
7. Discussion

Summary

- Motivation
 - Limited performance on text-only LLM agents
- Contribution
 - Propose a benchmark, VisualWebArena, with multimodal agents on realistic visually grounded web tasks
 - Evaluate state-of-the-art M/LLM-based autonomous agents



What is Autonomous Agents

Autonomous Agents for Web Browsing

Perform everyday tasks via human natural language commands

- Online shopping, booking tickets, searching information, etc
- Simple and direct tasks: add certain products into shopping carts
- Difficult tasks that requires reasoning and multiple steps.

Autonomous Agents for Web Browsing



“ Create an efficient itinerary to visit all of Pittsburgh's art museums with minimal driving distance starting from Schenley Park. Log the order in my “awesome-northeast-us-travel” repository ”

webarena.wikipedia.com

Wikipedia Pittsburgh museums

List of museums in Pittsburgh

This list of museums in **Pittsburgh, Pennsylvania** encompasses **museums** defined for this context as institutions (including **nonprofit organizations**, government entities, and private **businesses**) that collect and care for objects of cultural, artistic, scientific, or historical interest and make their collections or related exhibits available for public viewing. Also included are university and non-profit art galleries. Museums that exist only in cyberspace (i.e., **virtual museums**) are not included.

Wikimedia Commons has media related to **Museums in Pittsburgh**.

See also: *List of museums in Pennsylvania*

▼ Museums



Search for museums in Pittsburgh

webarena.openstreetmap.com

OpenStreetMap Edit History Export

Schenley Park, Pittsburgh, Allegheny County

The Andy Warhol Museum, 117, Sandusky Str

Car (OSRM)

Go

Reverse Directions

Directions

Distance: 7.1km. Time: 0:10.

1. Start on **Panther Hollow Road** 300m
2. Slight right onto unnamed road 160m



Search for each art museum on the Map

webarena.gitlab.com

README.md 158 B

Edit Replace

Travel in Northeast US

Pittsburgh

- + Miller Gallery at Carnegie Mellon University
- + American Jewish Museum
- + Carnegie Museum of Art



Record the optimized results to the repo

How to Evaluate an Autonomous Agent?

General Setup

- **Environment**

- Create a *realistic* and *reproducible* web environment
- The agent can analyze the given commands and current status and make a series of decisions to interact with the environment, getting closer to the final goal step by step.
- For example, Classifieds, Shopping, Reddit

- **Reward Function**

- Verify if the final goal is accomplished
- Different implementation for different tasks, e.g., direct match or GPT-assisted evaluation

Preliminary Work: WebArena

WEBARENA: A REALISTIC WEB ENVIRONMENT FOR BUILDING AUTONOMOUS AGENTS

Shuyan Zhou* Frank F. Xu*

Hao Zhu† Xuhui Zhou† Robert Lo† Abishek Sridhar† Xianyi Cheng
Tianyue Ou Yonatan Bisk Daniel Fried Uri Alon Graham Neubig

Carnegie Mellon University

{shuyanzh, fangzhex, gneubig}@cs.cmu.edu

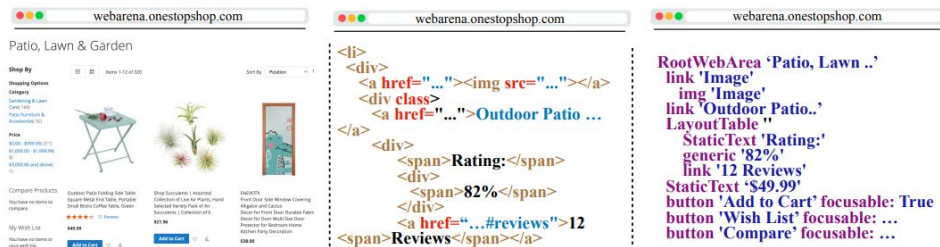
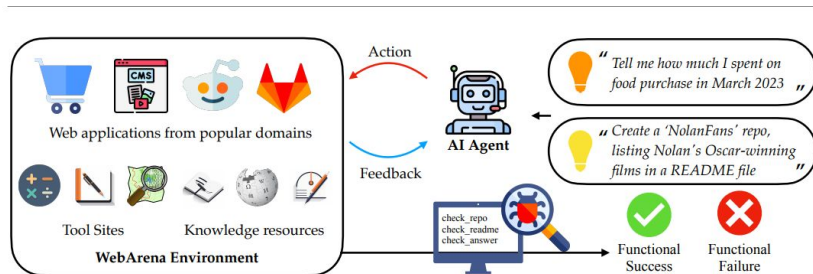


Figure 3: We design the observation to be the URL and the content of a web page, with options to represent the content as a screenshot (left), HTML DOM tree (middle), and accessibility tree (right). The content of the middle and right figures are trimmed to save space.

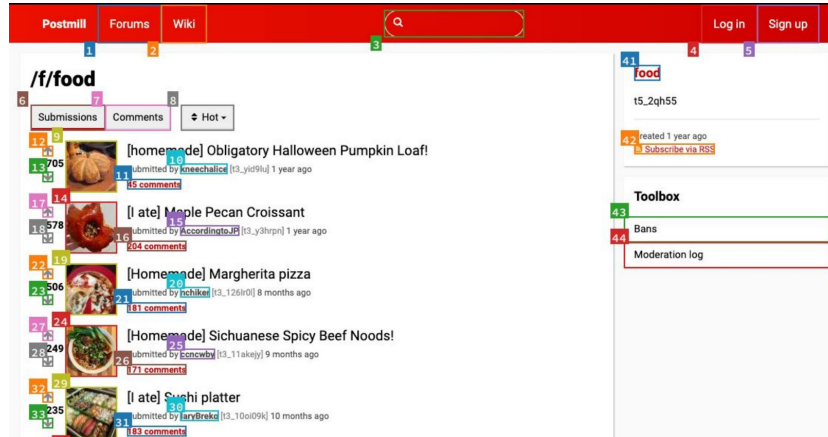
VisualWebArena: Necessity of Introducing Visual Input

Text-based

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

```
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: ...
```

Visual-based



The accessibility tree/html does not provide sufficient information to disentangle elements that are spatially close.

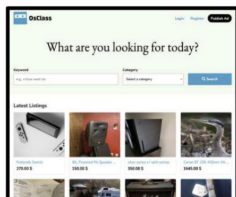
VisualWebArena: Website Setup

Webpage

Task specification



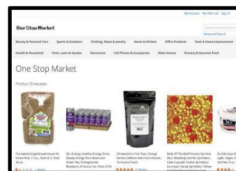
OsClass



"Help me make a post selling this item and navigate to it. Price it at \$10 cheaper than the most similar item on the site."



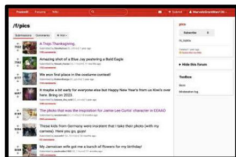
OneStopShop



"Buy the cheapest color photo printer and send it to Emily's place (as shown in the image)."



reddit

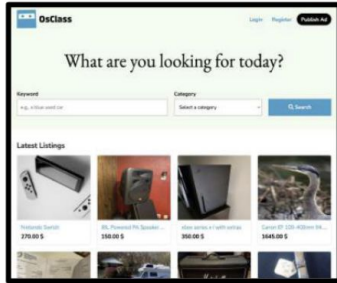


"Navigate to the comments section of the latest image post in the /f/Art subreddit that contains animals."

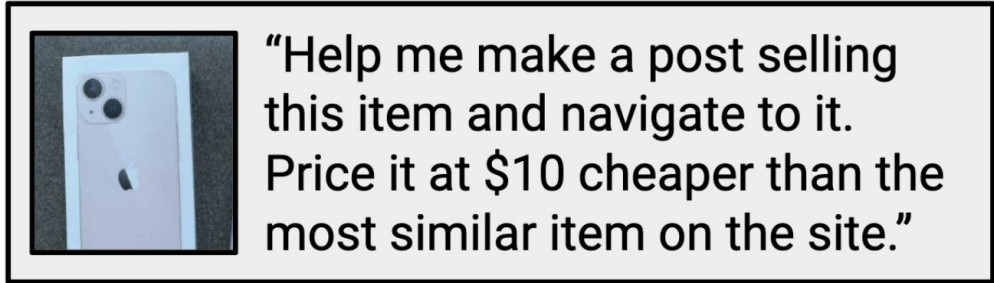
VisualWebArena: Website Setup - Classifieds



Webpage



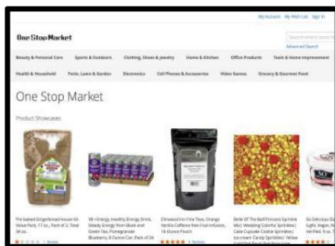
Task specification



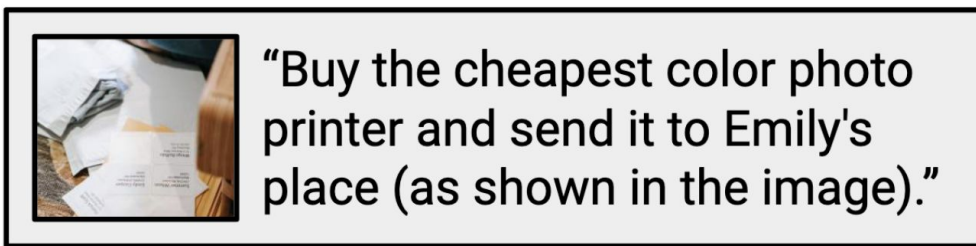
- Inspired by real-world marketplace, e.g., Facebook Marketplace
- Contains 65,955 listing
- User interaction: **posting, searching, commenting**

VisualWebArena: Website Setup - Shopping

Webpage



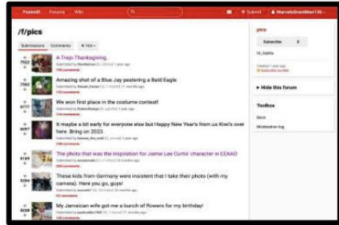
Task specification



- Contain product information and content from Amazon
- Contain ~90k products, including price, options, detailed product descriptions, images and reviews, spanning over 300 product categories

VisualWebArena: Website Setup - Reddit

Webpage



Task specification

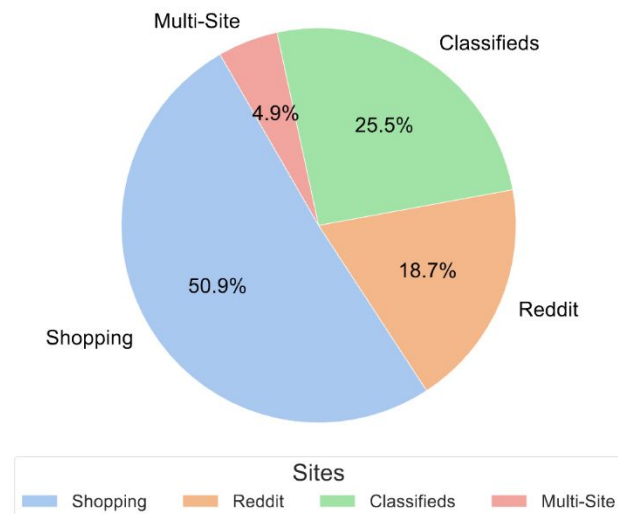
“Navigate to the comments section of the latest image post in the /f/Art subreddit that contains animals.”

- Contain 31,464 posts with a diverse set of images across different subreddits and forums, e.g., natural images, memes, consumer electronics, and charts

VisualWebArena: Data Curation

- Introduce a set of 910 new tasks for the 3 websites, e.g., Classifieds, Shopping, Reddit
- Hire 6 computer science graduate students to develop creative and realistic tasks
- 314 Template, e.g.,
 - “Find me the {{attribute}}{{item}}. It should be between {{range}}”
 - “Find me the cheapest red Toyota. It should be between \$3000 to \$6000.”

Distribution of Tasks Across Sites



VisualWebArena: Task Types

1. Information seeking tasks
2. Navigation and actions

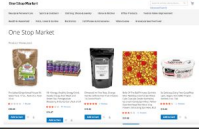



	Webpage / Input Image(s)	Example Intent	Reward Function $r(s, a)$ Implementation
1.		Buy the least expensive red blanket from the “Blankets & Throws” category.	<code>url="func:shopping_get_latest_order_url"</code> <code>must_include(â, { "B0983XCYK6", "Red" })</code>
2.		Add something like what the man is wearing to my wish list.	<code>url="/wishlist"</code> <code>locator(".wishlist .product-image-photo")</code> <code>eval_vqa(s, "Is this a polo shirt? (yes/no)", "yes")</code> <code>eval_vqa(s, "Is this shirt green? (yes/no)", "yes")</code>
2.		Create a post for each of these images in the most related forums.	<code>eval_fuzzy_image_match(s, a*)</code>
2.		Navigate to my listing of the white car and change the price to \$25000. Update the price in the description as well.	<code>url="/index.php?page=item&id=84144"</code> <code>must_include(â, "\$25000 OR \$25,000")</code> <code>must_exclude(â, "\$30000 OR \$30,000")</code>

Table 2: Various evaluation metrics to assign reward $r(s, a) \in R : S \times A \rightarrow \{0, 1\}$. Our execution-based reward primitives allow us to benchmark many diverse, realistic, and open-ended tasks.

Reward Functions Implementation

1. Information seeking tasks

Text functions

- `exact_match`
- `must_include`
- `fuzzy_match`
- `must_exclude`

Image functions

- `eval_vqa`
- `eval_fuzzy_image_match`

Reward Functions Implementation


1. Information seeking tasks

Text functions

- `exact_match`
- `must_include`
- `fuzzy_match`
- `must_exclude`

Image functions

- `eval_vqa`
- `eval_fuzzy_image_match`

Webpage / Input Image(s)	Example Intent	Reward Function $r(s, a)$ Implementation
	What is the ISIN of the company that occupies the largest portion in Warren Buffet's portfolio? Answer using the information from the Wikipedia site in the second tab.	<code>exact_match(\hat{a}, "US0378331005")</code>

Reward Functions Implementation

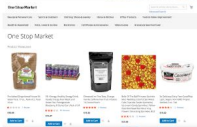
1. Information seeking tasks

Text functions

- `exact_match`
- `must_include`
- `fuzzy_match`
- `must_exclude`

Image functions

- `eval_vqa`
- `eval_fuzzy_image_match`

Webpage / Input Image(s)	Example Intent	Reward Function $r(s, a)$ Implementation
	Buy the least expensive red blanket from the “Blankets & Throws” category.	<code>url="func:shopping_get_latest_order_url"</code> <code>must_include(\hat{a}, { "B0983XCYK6", "Red" })</code>

Reward Functions Implementation

1. Information seeking tasks

Text functions

- exact_match
- must_include
- fuzzy_match
- must_exclude

Example intent:

Asking the user to add a comment describing an image

Image functions

- eval_vqa
- eval_fuzzy_image_match

Reward Functions Implementation


1. Information seeking tasks

Text functions

- exact_match
- must_include
- fuzzy_match
- **must_exclude**

Image functions

- eval_vqa
- eval_fuzzy_image_match

Webpage / Input Image(s)	Example Intent	Reward Function $r(s, a)$ Implementation
	Navigate to my listing of the white car and change the price to \$25000. Update the price in the description as well.	<code>url="/index.php?page=item&id=84144"</code> <code>must_include(\hat{a}, "\$25000 OR \$25,000")</code> <code>must_exclude(\hat{a}, "\$30000 OR \$30,000")</code>

Reward Functions Implementation


1. Information seeking tasks

Text functions

- exact_match
- must_include
- fuzzy_match
- must_exclude

Image functions

- eval_vqa
- eval_fuzzy_image_match

Webpage / Input Image(s)	Example Intent	Reward Function $r(s, a)$ Implementation
	Add something like what the man is wearing to my wish list.	<pre>url="/wishlist" locator(".wishlist .product-image-photo") eval_vqa(s, "Is this a polo shirt? (yes/no)", "yes") eval_vqa(s, "Is this shirt green? (yes/no)", "yes")</pre>

Reward Functions Implementation


1. Information seeking tasks

Text functions

- `exact_match`
- `must_include`
- `fuzzy_match`
- `must_exclude`

Image functions

- `eval_vqa`
- `eval_fuzzy_image_match` (with SSIM)

Webpage / Input Image(s)	Example Intent	Reward Function $r(s, a)$	Implementation
	Create a post for each of these images in the most related forums.	<code>eval_fuzzy_image_match(s, a*)</code>	

Reward Functions Implementation

1. Information seeking tasks

2. Navigation and actions

Locator:

- Describe the object on the page that should be examined (e.g., all img elements)
- Retrieve the corresponding image or text content



Reuse the metrics from information seeking tasks

Text functions

- `exact_match`
- `must_include`
- `fuzzy_match`
- `must_exclude`

Image functions

- `eval_vqa`
- `eval_fuzzy_image_match`

How to Formulate the Interaction Between an Agent and the Environment

Markov Decision Process

Markov decision process (MDP), also called a [stochastic dynamic program](#) or stochastic control problem, is a model for [sequential decision making](#)

Represented by a 4-tuple $\mathcal{E} = (S, A, \Omega, T)$

S set of states (the status of the whole browser)

A set of actions (the actions that the agent can perform)

Ω set of observations (the information that is sent to the agent)

T transition function $S \times A \rightarrow S$

(defines the state change after certain actions)

Action Space

Action Type <i>a</i>	Description
click [elem]	Click on element elem.
hover [elem]	Hover on element elem.
type [elem] [text]	Type text on element elem.
press [key_comb]	Press a key combination.
new_tab	Open a new tab.
tab_focus [index]	Focus on the i-th tab.
tab_close	Close current tab.
goto [url]	Open url.
go_back	Click the back button.
go_forward	Click the forward button.
scroll [up down]	Scroll up or down the page.
stop [answer]	End the task with an output.

Actions within one page

Actions on/between the pages

Observation - Set of Mark Visual Representaion

Previous work

- Website screen-shot
- HTML/DOM/Accessibility tree

[1744] link 'HP CB782A#ABA 640 Inkjet Fax Machine (Renewed)'
[1749] StaticText '\$279.49'
[1757] button 'Add to Cart'
[1760] button 'Add to Wish List'
[1761] button 'Add to Compare'

Set of Mark Representation

List the interactive elements and assign an ID. Mark each element with a bounding box and its corresponding ID number on the website screenshot.

Observation - Set of Mark Visual Representaion

The image shows a screenshot of a forum page with several numbered annotations (1-44) pointing to various interactive elements. The page has a red header with navigation links: Postmill (1), Forums (2), Wiki (3), a search bar (4), Log in (5), and Sign up (6). The main content area is titled "/f/food" and features a list of posts. Each post includes a submission count, a title, a submitter name, and a comment count. The posts are: 1. "[homemade] Obligatory Halloween Pumpkin Loaf!" (705 submissions, 15 comments) submitted by kneechalica. 2. "[I ate] Maple Pecan Croissant" (578 submissions, 204 comments) submitted by accordingtoJF. 3. "[Homemade] Margherita pizza" (506 submissions, 181 comments) submitted by hchiker. 4. "[Homemade] Sichuanese Spicy Beef Noods!" (249 submissions, 171 comments) submitted by conwby. 5. "[I ate] Sushi platter" (235 submissions, 183 comments) submitted by laryBrekd. On the right side, there is a sidebar with a "food" category (41), a post "t5_2qh55" (42) created 1 year ago with a "Subscribe via RSS" link (43), a "Toolbox" section (44) containing "Bans" and "Moderation log".

Webpage with SoM of Interactable Elements

Experiments

Compared Methods

- **Text-only LLM**
Accessibility Tree
- **Caption Augmented LLM**
Captions + Accessibility Tree
*Captions are provided by **BLIP-2** or **LLaVA***
- **Multi-modal LLM**
Image Screenshot + Captions + Accessibility Tree
- **Multi-modal LLM with SOM**
Image Screenshot + Captions + **SoM**

Experiments

Model Type	LLM Backbone	Visual Backbone	Inputs	Success Rate (\uparrow)			
				Classifieds	Reddit	Shopping	Overall
Text-only	LLaMA-2-70B			0.43%	1.43%	1.29%	1.10%
	Mixtral-8x7B			1.71%	2.86%	1.29%	1.76%
	Gemini-Pro	-	Acc. Tree	0.85%	0.95%	3.43%	2.20%
	GPT-3.5			0.43%	0.95%	3.65%	2.20%
	GPT-4			5.56%	4.76%	9.23%	7.25%
Caption-augmented	LLaMA-2-70B	BLIP-2-T5XL		0.00%	0.95%	0.86%	0.66%
	Mixtral-8x7B	BLIP-2-T5XL		1.28%	0.48%	2.79%	1.87%
	GPT-3.5	LLaVA-7B		1.28%	1.43%	4.08%	2.75%
	GPT-3.5	BLIP-2-T5XL	Acc. Tree + Caps	0.85%	1.43%	4.72%	2.97%
	Gemini-Pro	BLIP-2-T5XL		1.71%	1.43%	6.01%	3.85%
	GPT-4	BLIP-2-T5XL		8.55%	8.57%	16.74%	12.75%
Multimodal	IDEFICS-80B-Instruct			0.43%	0.95%	0.86%	0.77%
	CogVLM			0.00%	0.48%	0.43%	0.33%
	Gemini-Pro		Image + Caps + Acc. Tree	3.42%	4.29%	8.15%	6.04%
	GPT-4V			8.12%	12.38%	19.74%	15.05%
Multimodal (SoM)	IDEFICS-80B-Instruct			0.85%	0.95%	1.07%	0.99%
	CogVLM			0.00%	0.48%	0.43%	0.33%
	Gemini-Pro		Image + Caps + SoM	3.42%	3.81%	7.73%	5.71%
	GPT-4V			9.83%	17.14%	19.31%	16.37%
Human Performance	-	-	Webpage	91.07%	87.10%	88.39%	88.70%

Table 3: Success rates of baseline LLM and VLM agents on VisualWebArena.

Experiments

Model Type	LLM Backbone	Visual Backbone	Inputs	Success Rate (\uparrow)			
				Classifieds	Reddit	Shopping	Overall
Text-only	LLaMA-2-70B			0.43%	1.43%	1.29%	1.10%
	Mixtral-8x7B			1.71%	2.86%	1.29%	1.76%
	Gemini-Pro	-	Acc. Tree	0.85%	0.95%	3.43%	2.20%
	GPT-3.5			0.43%	0.95%	3.65%	2.20%
	GPT-4			5.56%	4.76%	9.23%	7.25%
Caption-augmented	LLaMA-2-70B	BLIP-2-T5XL		0.00%	0.95%	0.86%	0.66%
	Mixtral-8x7B	BLIP-2-T5XL		1.28%	0.48%	2.79%	1.87%
	GPT-3.5	LLaVA-7B	Acc. Tree + Caps	1.28%	1.43%	4.08%	2.75%
	GPT-3.5	BLIP-2-T5XL		0.85%	1.43%	4.72%	2.97%
	Gemini-Pro	BLIP-2-T5XL		1.71%	1.43%	6.01%	3.85%
	GPT-4	BLIP-2-T5XL		8.55%	8.57%	16.74%	12.75%
Multimodal	IDEFICS-80B-Instruct			0.43%	0.95%	0.86%	0.77%
	CogVLM		Image + Caps + Acc. Tree	0.00%	0.48%	0.43%	0.33%
	Gemini-Pro			3.42%	4.29%	8.15%	6.04%
	GPT-4V			8.12%	12.38%	19.74%	15.05%
Multimodal (SoM)	IDEFICS-80B-Instruct			0.85%	0.95%	1.07%	0.99%
	CogVLM		Image + Caps + SoM	0.00%	0.48%	0.43%	0.33%
	Gemini-Pro			3.42%	3.81%	7.73%	5.71%
	GPT-4V			9.83%	17.14%	19.31%	16.37%
Human Performance	-	-	Webpage	91.07%	87.10%	88.39%	88.70%

Table 3: Success rates of baseline LLM and VLM agents on VisualWebArena.

Experiments

Model Type	LLM Backbone	Visual Backbone	Inputs	Success Rate (\uparrow)			
				Classifieds	Reddit	Shopping	Overall
Text-only	LLaMA-2-70B		Acc. Tree	0.43%	1.43%	1.29%	1.10%
	Mixtral-8x7B			1.71%	2.86%	1.29%	1.76%
	Gemini-Pro	-		0.85%	0.95%	3.43%	2.20%
	GPT-3.5			0.43%	0.95%	3.65%	2.20%
	GPT-4			5.56%	4.76%	9.23%	7.25%
Caption-augmented	LLaMA-2-70B	BLIP-2-T5XL	Acc. Tree + Caps	0.00%	0.95%	0.86%	0.66%
	Mixtral-8x7B	BLIP-2-T5XL		1.28%	0.48%	2.79%	1.87%
	GPT-3.5	LLaVA-7B		1.28%	1.43%	4.08%	2.75%
	GPT-3.5	BLIP-2-T5XL		0.85%	1.43%	4.72%	2.97%
	Gemini-Pro	BLIP-2-T5XL		1.71%	1.43%	6.01%	3.85%
	GPT-4	BLIP-2-T5XL	8.55%	8.57%	16.74%	12.75%	
Multimodal	IDEFICS-80B-Instruct		Image + Caps + Acc. Tree	0.43%	0.95%	0.86%	0.77%
	CogVLM			0.00%	0.48%	0.43%	0.33%
	Gemini-Pro			3.42%	4.29%	8.15%	6.04%
	GPT-4V			8.12%	12.38%	19.74%	15.05%
Multimodal (SoM)	IDEFICS-80B-Instruct		Image + Caps + SoM	0.85%	0.95%	1.07%	0.99%
	CogVLM			0.00%	0.48%	0.43%	0.33%
	Gemini-Pro			3.42%	3.81%	7.73%	5.71%
	GPT-4V			9.83%	17.14%	19.31%	16.37%
Human Performance	-	-	Webpage	91.07%	87.10%	88.39%	88.70%




Table 3: Success rates of baseline LLM and VLM agents on VisualWebArena.

Experiments

Model Type	LLM Backbone	Visual Backbone	Inputs	Success Rate (\uparrow)			
				Classifieds	Reddit	Shopping	Overall
Text-only	LLaMA-2-70B			0.43%	1.43%	1.29%	1.10%
	Mixtral-8x7B			1.71%	2.86%	1.29%	1.76%
	Gemini-Pro	-	Acc. Tree	0.85%	0.95%	3.43%	2.20%
	GPT-3.5			0.43%	0.95%	3.65%	2.20%
	GPT-4			5.56%	4.76%	9.23%	7.25%
Caption-augmented	LLaMA-2-70B	BLIP-2-T5XL		0.00%	0.95%	0.86%	0.66%
	Mixtral-8x7B	BLIP-2-T5XL		1.28%	0.48%	2.79%	1.87%
	GPT-3.5	LLaVA-7B	Acc. Tree + Caps	1.28%	1.43%	4.08%	2.75%
	GPT-3.5	BLIP-2-T5XL		0.85%	1.43%	4.72%	2.97%
	Gemini-Pro	BLIP-2-T5XL		1.71%	1.43%	6.01%	3.85%
	GPT-4	BLIP-2-T5XL		8.55%	8.57%	16.74%	12.75%
Multimodal	IDEFICS-80B-Instruct			0.43%	0.95%	0.86%	0.77%
	CogVLM		Image + Caps + Acc. Tree	0.00%	0.48%	0.43%	0.33%
	Gemini-Pro			3.42%	4.29%	8.15%	6.04%
	GPT-4V			8.12%	12.38%	19.74%	15.05%
Multimodal (SoM)	IDEFICS-80B-Instruct			0.85%	0.95%	1.07%	0.99%
	CogVLM		Image + Caps + SoM	0.00%	0.48%	0.43%	0.33%
	Gemini-Pro			3.42%	3.81%	7.73%	5.71%
	GPT-4V			9.83%	17.14%	19.31%	16.37%
Human Performance	-	-	Webpage	91.07%	87.10%	88.39%	88.70%

Table 3: Success rates of baseline LLM and VLM agents on VisualWebArena.

Experiments

Multimodal (SoM)	IDEFICS-80B-Instruct		0.85%	0.95%	1.07%	0.99%	
	CogVLM		0.00%	0.48%	0.43%	0.33%	
	Gemini-Pro	Image + Caps + SoM	3.42%	3.81%	7.73%	5.71%	
	GPT-4V		9.83%	17.14%	19.31%	16.37%	
Human Performance	-	-	Webpage	91.07%	87.10%	88.39%	88.70%

Agent Backbone	Model Type	Success Rate (↑)			
		Classifieds	Reddit	Shopping	Overall
Llama-3-70B-Instruct	Caption-augmented	7.69%	5.24%	12.88%	9.78%
Gemini-Flash-1.5	Image + Caps + SoM	3.85%	4.76%	8.80%	6.59%
Gemini-Pro-1.5	Image + Caps + SoM	5.98%	12.86%	14.59%	11.98%
GPT-4o	Image + Caps + SoM	20.51%	16.67%	20.82%	19.78%

Table 5: Success rates of recent LLM and VLM agents on VisualWebArena.

Experiments - Example of Execution Trajectory

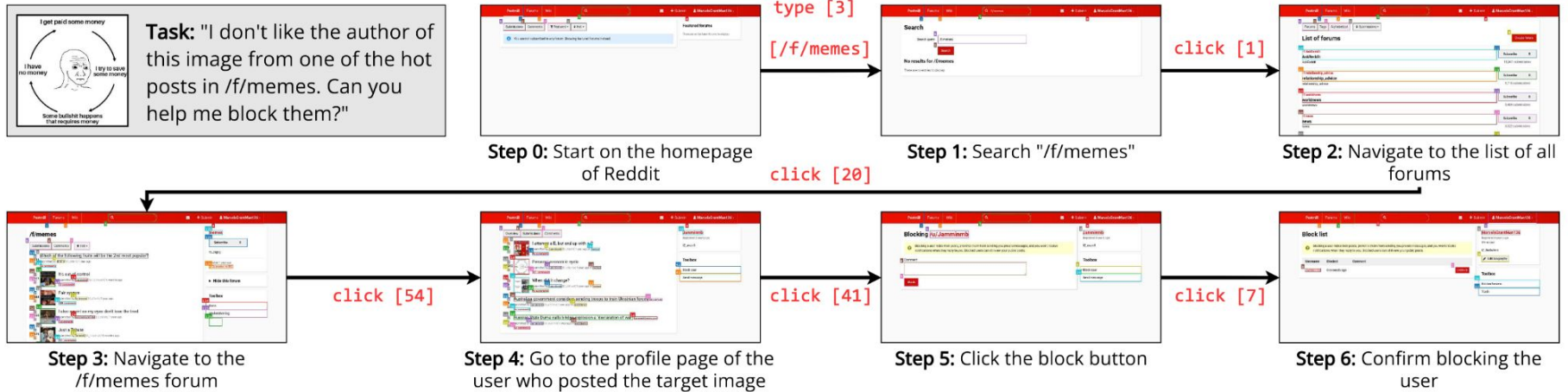


Figure 3: Successful execution trajectory of the GPT-4V + SoM agent on the task for blocking a user that posted a certain picture. The text in red represents the actions output by the agent.

Failure Cases

Failure Over Longer Horizons

- Correctly perform a task but undo it, leading to failure
- For example, adding a product to the wishlist, but remove it later

Giving Up Too Early

- LLM think the given task is not achievable.
- For example, fail to see some elements because the page need to be scrolled down.
- Even human can fail because of this reason.

Getting Stuck in Loops

- Goes back and repeats from the previous steps
- For example, keep switching between different tabs to compare products

Conclusion

Conclusion

- Create a visual-conditioned environment for web agent.
- Propose a new VLM agent inspired by “Set of Marks prompting.”
- Benchmark the open-source and commercial models.

Key Findings

- The success rates of all methods, including the most advanced commercial models, are still unsatisfactory.
- Incorporating visual signals and representations is crucial for web agents.

Limitations

- LLM/MLLM primarily function as controllers and schedulers. There is still much potential for improving the whole system, e.g., the prompt design or methods to feed information into LLM/MLLM.
- Some MLLMs designed for web/UI agents are not included in the comparison and analysis. For example, CogAgent modify the architecture based on CogVLM to support high-resolution image and add more SFT data on UI operations.

Discussion

- Is SSIM a good evaluation metric for `eval_fuzzy_image_match`?
- VisualWebArena largely repeats WebArena. How would you evaluate VisualWebArena as a reviewer?