Admin

- Participation component
  - **Slido:** Will provide a QR code on presentation, **submit at least 2 questions for rest of the semester!**
  - **Ed**: Will provide an (AI) summary of class, along with discussion items, right after class. **submit at least 2 (non-trivial!) posts**

- Will have comments out, TAs + me

- **Synthesis:**
  - Nov 24 – Synthesis in-person discussion session
  - We will also do one end of class if time permits after presentations
    - Note that you will have a lot more time for final presentations!

Georgia Tech

Mind2Web → MolmoWeb

Deng et al.
NeurIPS 2023

Gupta et al.
arXiv 2026

Jianzhe Wei, Mehdi Heydari, Ankit Roy

Georgia Tech

Outline

- Problem Statement
- Related Works
- Mind2Web
  - Approach
  - Experiments & Results
- MolmoWeb
  - Approach
  - Experiments & Results
- What persisted and what changed?
- Limitations and Implications
- Discussion Questions

Georgia Tech

Problem Statement

- **Goal**: Build a general-purpose AI agent capable of completing complex tasks on real-world websites using natural-language instruction
- Challenges:

Complex, dynamic webpage environments

Diverse tasks and website interfaces

Understanding high-level user intentions

Generalizing to unseen websites and domains

Georgia Tech

Related Works

- **WebShop [1]**
  - Introduced a simulated e-commerce environment with 1.18 million real products and over 12,000 natural-language shopping instructions
  - **Limitation**: Restricted to shopping tasks in a simulated website; limited generalization to other domains
- **WebArena [2]**
  - Realistic, fully functional websites for evaluating long-horizon web agents and their task completion
  - **Limitation**: Uses a controlled set of replicated websites rather than the unrestricted live web
- **WebVoyager [3]**
  - Developed a GPT-4V-based web agent that uses screenshots with labeled interactive elements to execute tasks on live websites
  - **Limitation**: Relies on a proprietary multimodal model and annotated screenshots; evaluated across 15 websites

[1] Yao, S., et al., 2022 [1] Zhou, S., et al., 2024 [3] He, H., et al., 2024

Mind2Web Dataset

2,350
tasks

137
websites

31
domains

- Each dataset example contained:
  - Task description: high level instruction describing what the user wants
  - Action sequence: sequence of actions (target element, operation) needed to complete the task
    - CLICK, TYPE, SELECT OPTION
  - Webpage snapshots:
    - HTML, DOM snapshot, network traffic, and interaction trace
- The agent receives the current webpage, task description, and previous actions to predict the next action

Georgia Tech

Dataset Collection

01

Site Selection

Representative sites across domains

→

02

Propose tasks

MTurk Annotators write high-level goals

→

03

Demonstrate

Complete tasks and record every action

→

04

Verify

Review goals, actions, and results

Georgia Tech

Dataset Collection

Task Description:
Show me the reviews for the auto repair business closest to 10002.

Action Sequence:

	Target Element	Operation
1.	[searchbox] *Find*	**TYPE:** ***auto repair***
2.	[button] *Auto Repair*	**CLICK**
3.	[textbox] *Near*	**TYPE:** ***10002***
4.	[button] *10002*	**CLICK**
5.	**[button] *Search***	**CLICK**
6.	[switch] *Show BBB Accredited only*	**CLICK**
7.	[svg]	**CLICK**
8.	[button] *Sort By*	**CLICK**
9.	**[link] *Fast Lane 24 Hour Auto Repair***	**CLICK**
10.	[link] *Read Reviews*	**CLICK**

Webpage Snapshots:




```
<input name="find_text"
type="search">
```




```
<em>Auto Repair</em>
```




```
<button>Search</button>
```




```
<button>Show BBB Accredited
only</button>
```




```
<span>Fast Lane 24 Hour Auto
Repair</span>
```




```
<a href="link:XXX">Read
Reviews</a>
```

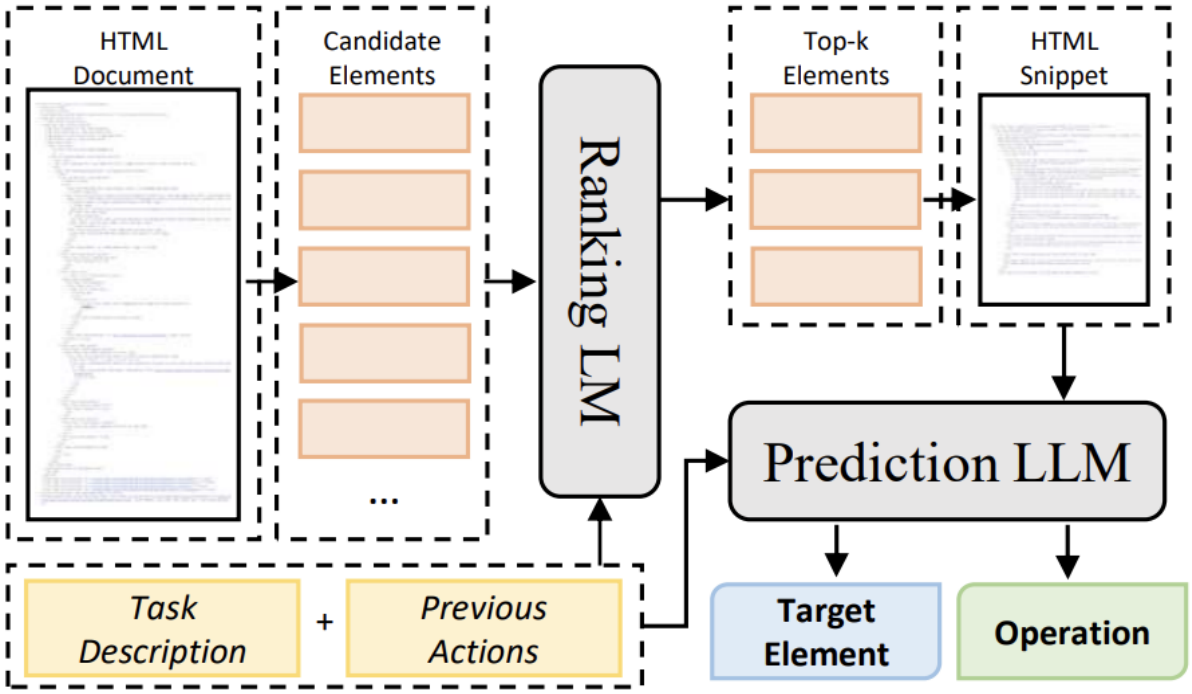
Deng et al., Mind2Web (NeurIPS 2023), Figure 2

Georgia Tech

MindAct Architecture

- **Bottleneck**: Real-world HTML webpages can contain **thousands of elements**, making it expensive or impractical to feed the entire page into an LLM
- MindAct uses a **two-stage** architecture:
  - **Stage 1**: A smaller language model identifies the most relevant webpage elements
  - **Stage 2**: A larger language model selects the correct element and predicts the corresponding operation

- **Input**:
  - User task description
  - Current webpage HTML
  - History of previous actions
- **Output**:
  - Target DOM element
  - Operation



Deng et al., Mind2Web (NeurIPS 2023), Figure 3

Georgia Tech

MindAct (Candidate Generation)

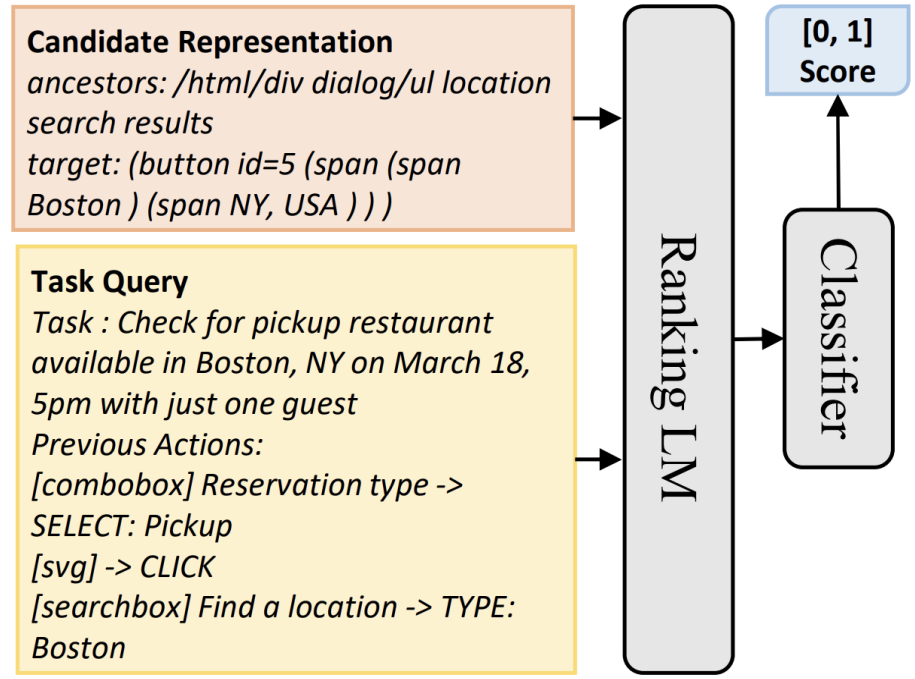
- Goal: Rank DOM elements based on their relevance to the current task
- Uses **DeBERTa-v3-base**, an encoder-only LM with ~86 million parameters
- Processes the task query and element representation and produces a relevance score for **each element**
- Failure to include the correct target element among the candidates can lead to cascading errors

Training

Positive examples: Correct target elements
Negative examples: Randomly sampled incorrect elements
Uses sigmoid activation and binary cross-entropy loss

Inference

Scores the webpage elements
Selects the top 50 candidates
Passes them to the larger language model



Deng et al., Mind2Web (NeurIPS 2023), Figure 4

Georgia Tech

MindAct (Action Prediction)

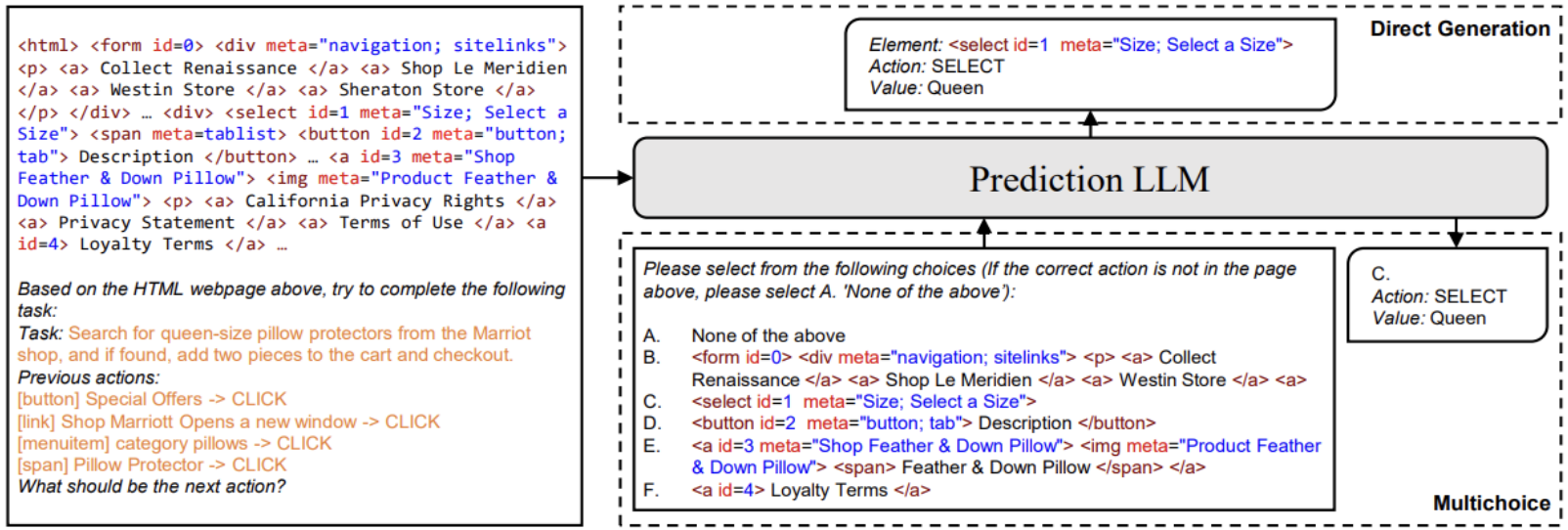
- Uses an LLM (Flan-T5, GPT) to select the correct candidate and predict the operation

Direct generation

LLM directly generates the HTML element and corresponding action

Multiple-choice question answering

Present five candidates at a time
LLM selects the element most relevant to the next action, or chooses "None of the above"
LLM repeatedly selects promising candidates until one element remains



Deng et al., Mind2Web (NeurIPS 2023), Figure 5

Georgia Tech

Evaluation Setup

Cross-Task

Tests different tasks on websites encountered during training

Model has prior exposure to the website's structure

252 test tasks from 69 websites

Easiest of the three settings

Cross-Website

Tests tasks on completely unseen websites

Websites belong to domains represented in training

177 test tasks from 10 held-out websites

Evaluates transfer to new website layouts

Cross-Domain

Tests tasks from entirely unseen top-level domains

Information and Service domains were held out

912 test tasks from 73 websites

Evaluates generalization beyond previously observed domains

Georgia Tech

Results

	Cross-Task				Cross-Website				Cross-Domain			
	Ele. Acc	Op. F1	Step SR	SR	Ele. Acc	Op. F1	Step SR	SR	Ele. Acc	Op. F1	Step SR	SR
Classification	26.8	–	–	–	21.6	–	–	–	24.5	–	–	–
Generation	20.2	52.0	17.5	0.0	13.9	44.7	11.0	0.0	14.2	44.7	11.9	0.4
MINDACT												
w/ Flan-$T5_B$	43.6	76.8	41.0	4.0	32.1	**67.6**	29.5	1.7	33.9	**67.3**	31.6	1.6
w/ Flan-$T5_L$	53.4	**75.7**	50.3	**7.1**	39.2	67.1	35.3	1.1	39.7	67.2	37.3	2.7
w/ Flan-$T5_{XL}$	**55.1**	**75.7**	**52.0**	5.2	**42.0**	65.2	**38.9**	**5.1**	**42.1**	66.5	**39.6**	**2.9**
w/ GPT-3.5	20.3	56.6	17.4	0.8	19.3	48.8	16.2	0.6	21.6	52.8	18.6	1.0
w/ GPT-4*	41.6	60.6	36.2	2.0	35.8	51.1	30.1	2.0	37.1	46.5	26.4	2.0

Deng et al., Mind2Web (NeurIPS 2023), Table 2

- **Element Accuracy**: % of steps where the selected webpage element matches an acceptable ground-truth element.
- **Operation F1**: Token-level F1 comparing the predicted operation and value to the ground truth. For CLICK, this reduces to accuracy; for TYPE and SELECT, the input value is also evaluated.
- **Step Success Rate**: % of steps where both the target element and operation are correct.
- **Task Success Rate**: % of tasks where every step is successful. A single incorrect step causes the entire task to fail.

Georgia Tech

Takeaways

	Cross-Task				Cross-Website				Cross-Domain			
	Ele. Acc	Op. F1	Step SR	SR	Ele. Acc	Op. F1	Step SR	SR	Ele. Acc	Op. F1	Step SR	SR
Classification	26.8	–	–	–	21.6	–	–	–	24.5	–	–	–
Generation	20.2	52.0	17.5	0.0	13.9	44.7	11.0	0.0	14.2	44.7	11.9	0.4
MINDACT												
w/ Flan-T5$_B$	43.6	76.8	41.0	4.0	32.1	**67.6**	29.5	1.7	33.9	**67.3**	31.6	1.6
w/ Flan-T5$_L$	53.4	**75.7**	50.3	**7.1**	39.2	67.1	35.3	1.1	39.7	67.2	37.3	2.7
w/ Flan-T5$_{XL}$	**55.1**	**75.7**	**52.0**	5.2	**42.0**	65.2	**38.9**	**5.1**	**42.1**	66.5	**39.6**	**2.9**
w/ GPT-3.5	20.3	56.6	17.4	0.8	19.3	48.8	16.2	0.6	21.6	52.8	18.6	1.0
w/ GPT-4*	41.6	60.6	36.2	2.0	35.8	51.1	30.1	2.0	37.1	46.5	26.4	2.0

Deng et al., Mind2Web (NeurIPS 2023), Table 2

- **Error propagation is a major bottleneck**: Even with 52% step success, only 5.2% of full tasks succeed
- **Element selection is more challenging than action prediction**: Higher Operation F1 than Element Accuracy suggests that identifying *where* to act is harder than determining what action to perform.
- **Larger models alone cannot solve web navigation**: Full-task success remains extremely low
- **Generalization remains difficult**: Performance drops on unseen websites and domains

Georgia Tech

Limitations

Text-Only Understanding

MindAct uses textual HTML representations rather than actual webpage screenshots

Low Task Success Rate

Full-task success is extremely low despite moderate step-level performance

Candidate Ranking Bottleneck

Correct element must survive HTML preprocessing and candidate ranking

Offline Evaluation

Tested using cached webpage states rather than live websites

Bridge: visual web agents

1. End-to-end execution on long-horizon tasks

Take actions, receive feedback, and continue until the task is complete.

2. Visual grounding

Identify objects from language descriptions and precisely localize them in images.

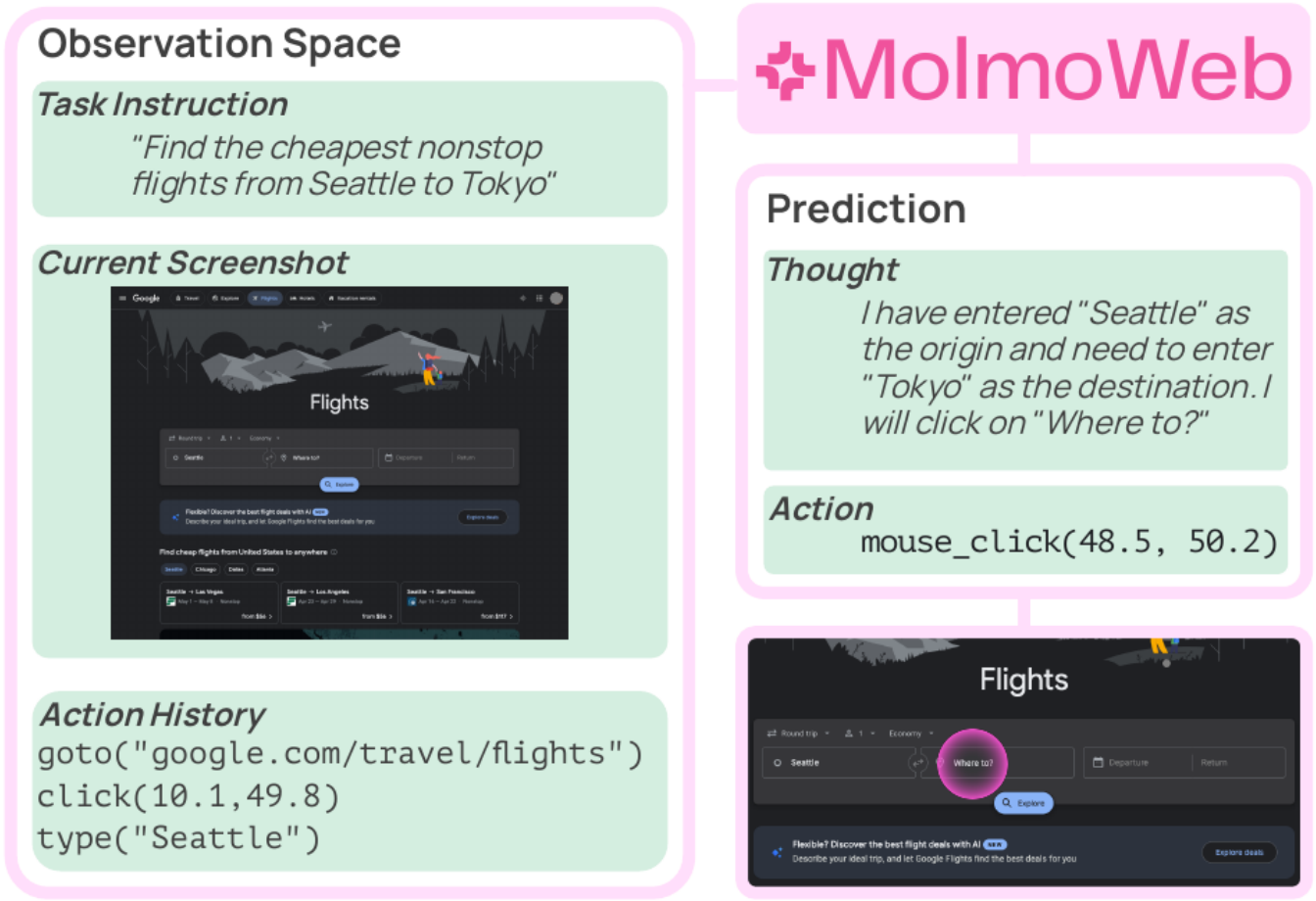
MolmoWeb: Can we build an open VLM web agent and an open training recipe?

Idea: Collect demonstrations and perception data, then train with SFT.

Deng et al., 2023, §4.2. Gupta et al., 2026, §§1, 3–4.

Georgia Tech

Approach: one visual action policy



Input

Screenshot, task, URL / title, last 10 actions

Prediction

JSON thought + one action

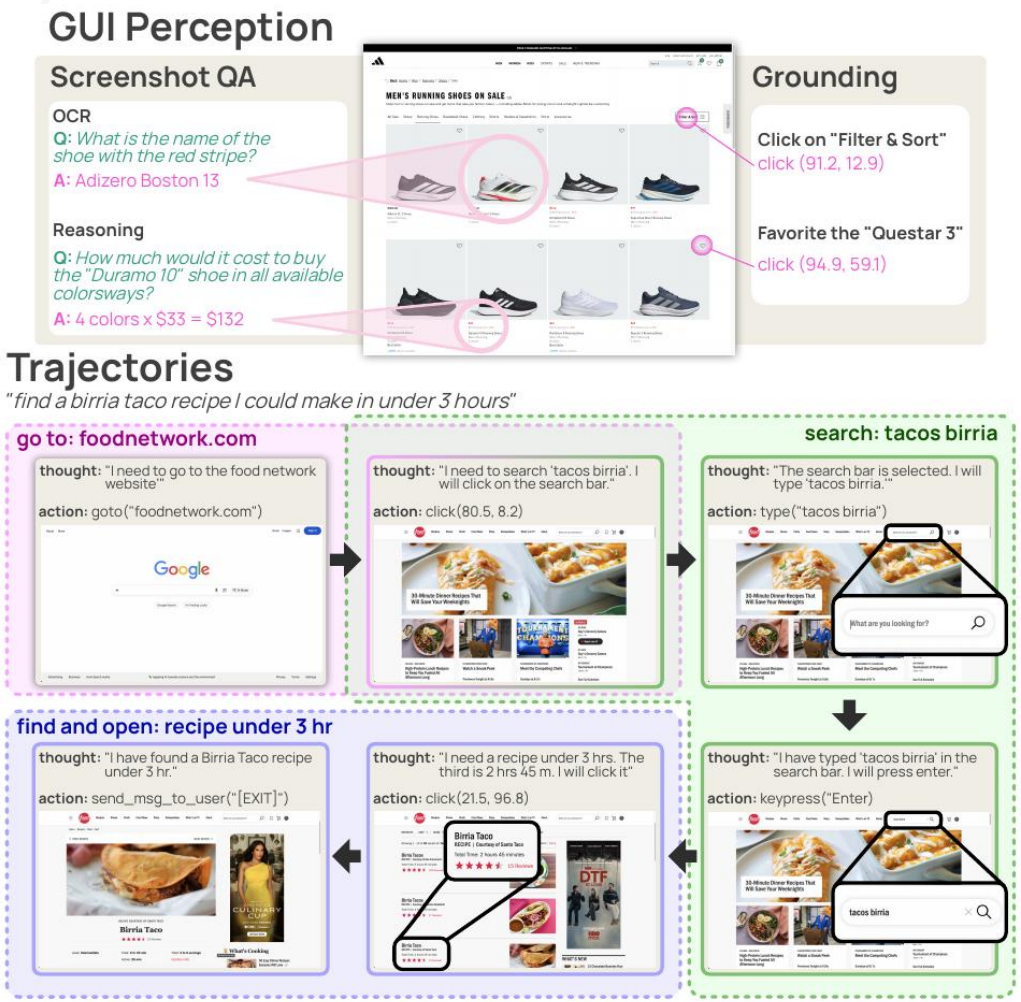
Execution

Coordinates in [0, 100] become viewport pixels

Gupta et al., 2026, Fig. 1 and §§3.1–3.2

Georgia Tech

Data: what a web agent needs to learn



1. Complete tasks

Demonstrate how to reach a user's goal

2. Atomic skills

Reuse task segments, such as search and filtering

3. GUI perception

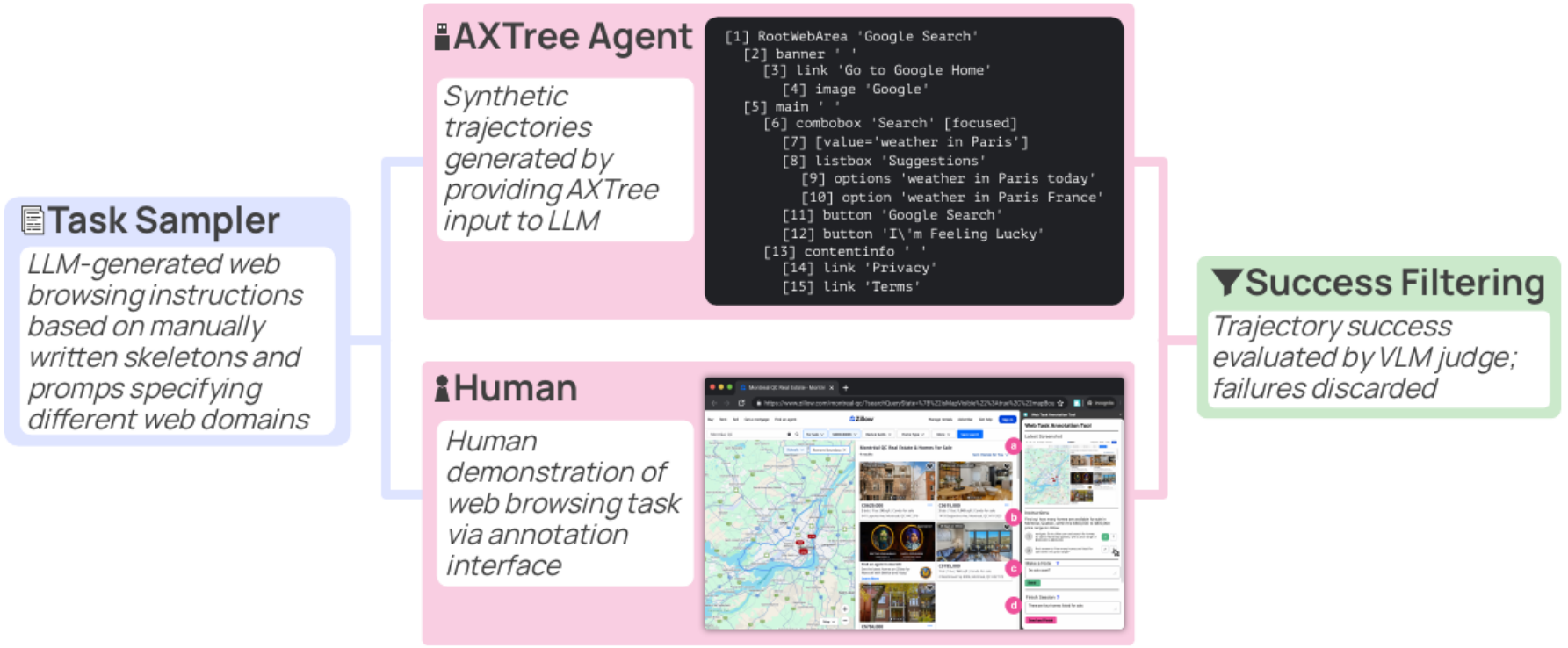
Read the page and locate controls

Gupta et al., 2026, Fig. 2 and §§2.1–2.3.

Georgia Tech

Full trajectory: an AxTree agent teacher

AxTree Agent | Human Trajectory | Multi-Agent | Node Traversal



Teacher

Input: AxTree (cleaner than DOM)
Output: element-ID actions in AxTree

Student

Input: screenshot
Output: coordinates for viewport actions like click

Convert element-ID actions to coordinates and pair them with screenshots

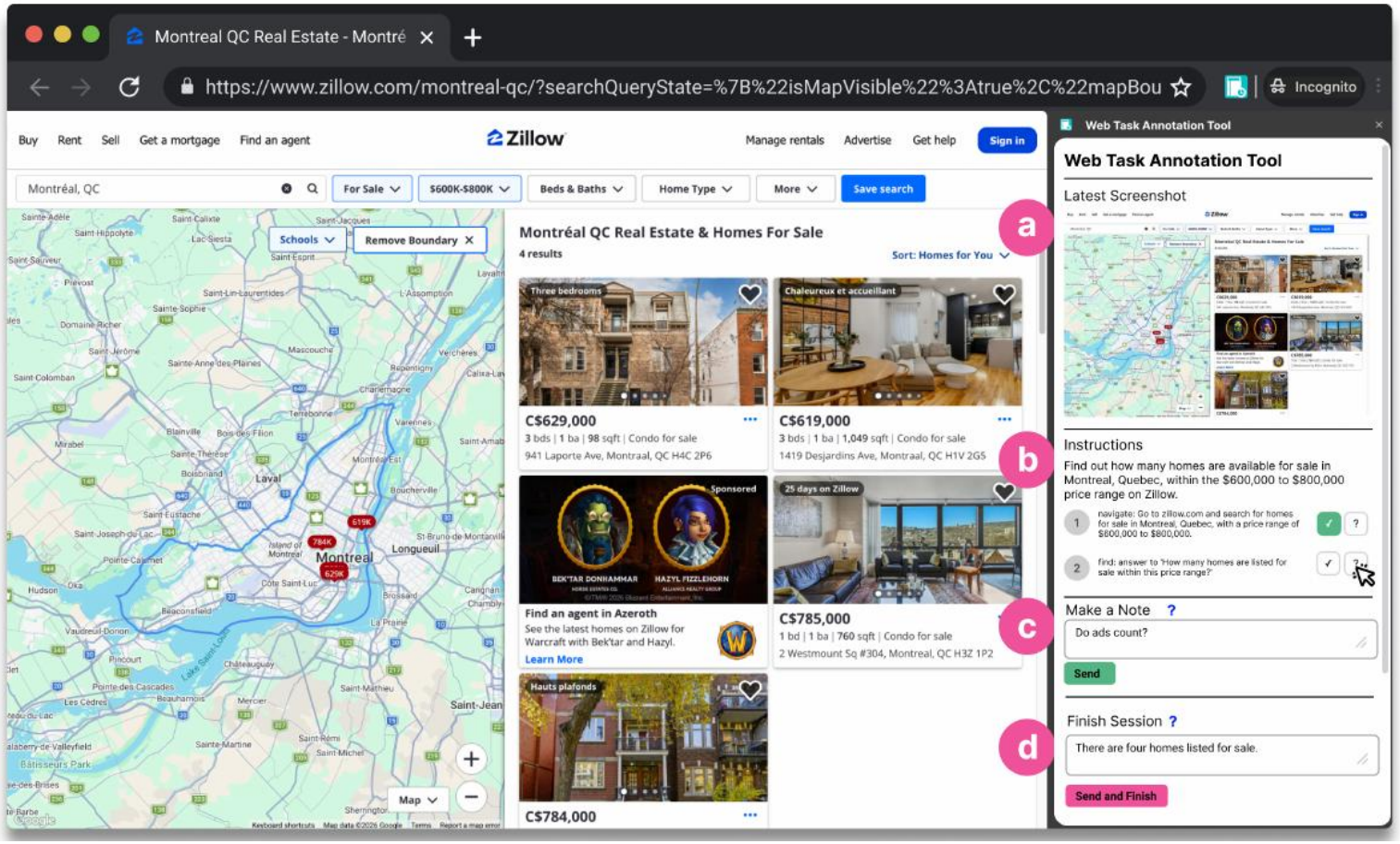
AXTree Agent: Gemini-3-Flash-Preview – teacher with high success rate

Gupta et al., 2026, Fig. 3 and §2.1

Georgia Tech

Full trajectory: human task demonstrations

AxTree Agent | **Human Trajectory** | Multi-Agent | Node Traversal



Human trajectory

Chrome extension records events and screenshots

Subtask boundaries

Workers check off each skill

Human review

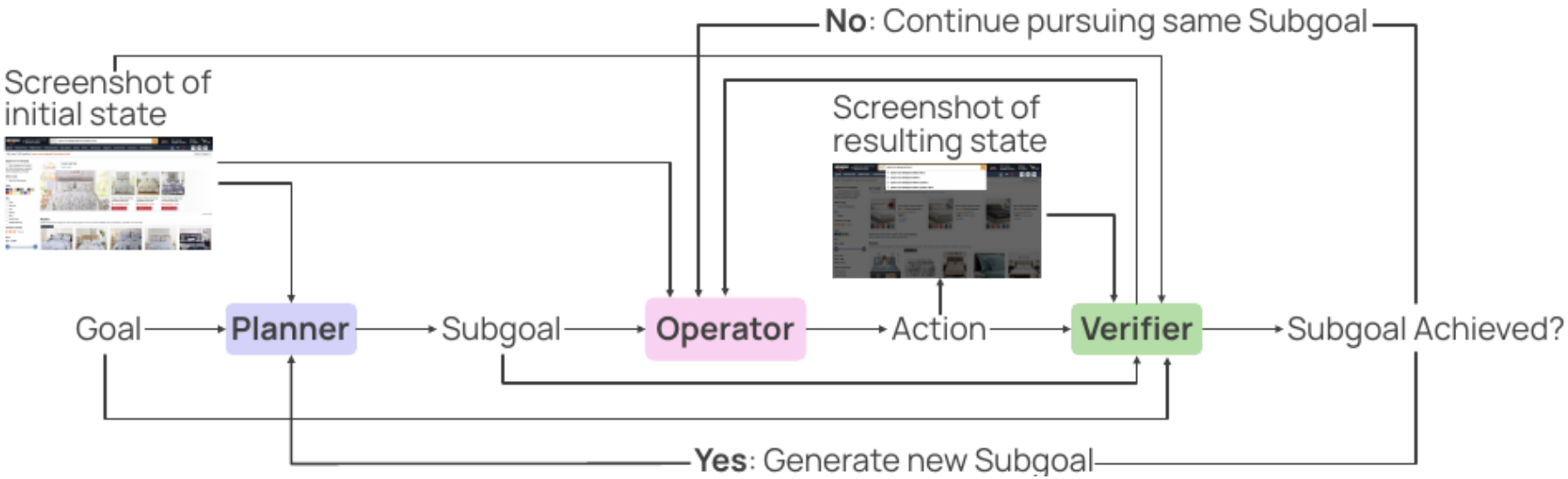
Revise or recollect failures

Gupta et al., 2026, Fig. 7, §2.1 and App. B

Georgia Tech

Full trajectory: multi-agent generation

AxTree Agent | Human Trajectory | **Multi-Agent** | Node Traversal



Planner	Operator	Verifier
Next subgoal	One browser action	Recent screenshots
Gemini-2.5-Flash	Gemini AxTree agent	GPT-4o

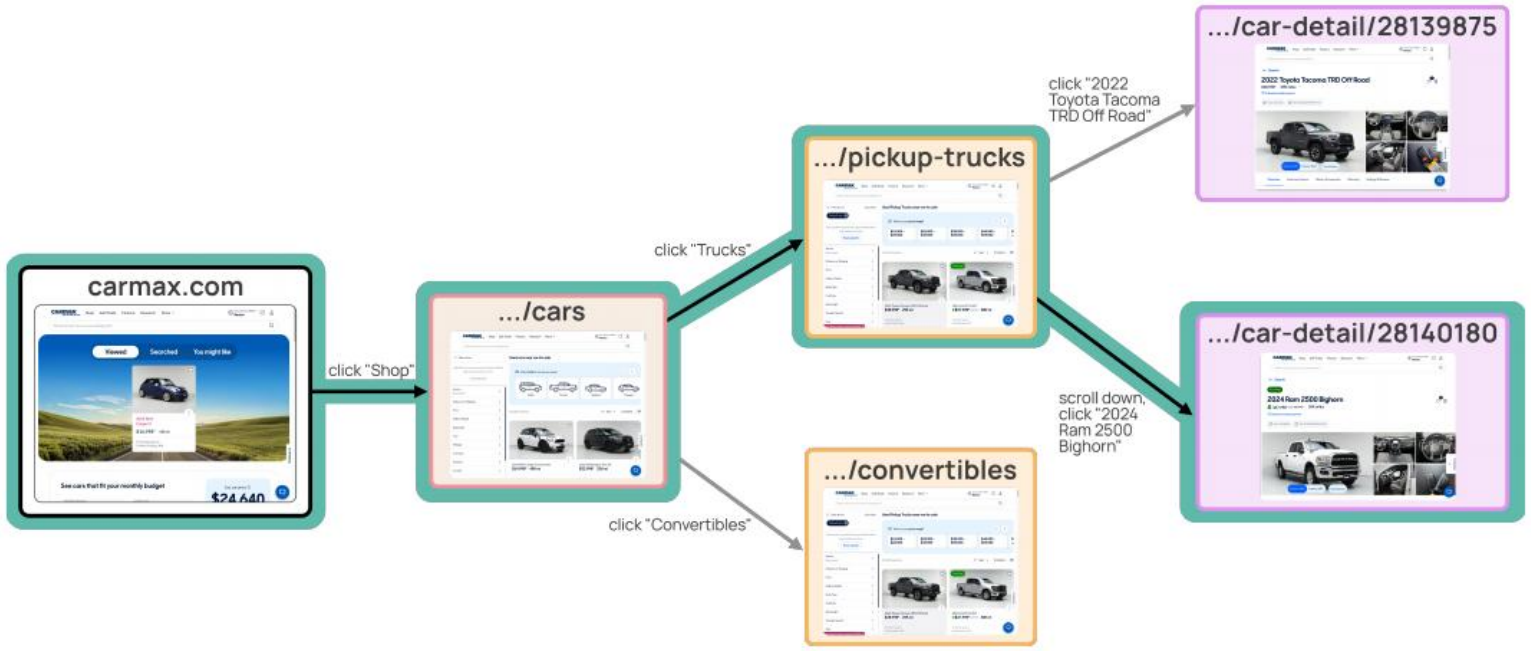
Higher task success rate: 78.5 vs 74.4 (single agent); higher quality potentially

Gupta et al., 2026, Fig. 4 and §2.1

Georgia Tech

Full trajectory: node traversal

AxTree Agent | Human Trajectory | Multi-Agent | **Node Traversal**



Plan a URL route

Root to Leaf:
Explore the website graph

Replay the route

URLs to Screenshots:
Deterministic click / scroll

Write a matching goal

LLM labels the finished path

Complement LLM and human task-complete trajectories

Gupta et al., 2026, Fig. 5 and §2.1

Georgia Tech

Atomic web skills

Complete tasks | **Atomic skills** | GUI perception

Web-use tasks consists of shared and reusable subtasks (skills):

E.g., 'search' for shopping and browsing; 'find' for flight booking and car renting

11 skills in paper: go_to, search, find, fill_form, filter, cart, navigate, …

Atomic skill can be multi-step trajectories:

A search can involve clicking a field, typing a query, and pressing Enter.

Source	Construction
Human demonstrations	Cut at annotated subtask boundaries
AxTree agent	Generate 'form-filling' and 'find-and-open' tasks

Gupta et al., 2026, §2.2 and Tables 1–2.

Georgia Tech

Visual perception for web use

Complete tasks | Atomic skills | **GUI perception**

The policy acts on what it can read and locate in a screenshot.

GUI Perception

Screenshot QA

OCR

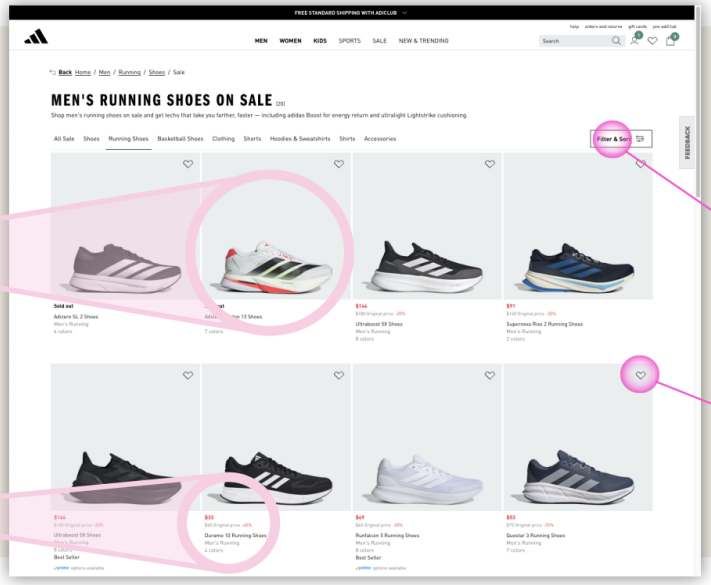
Q: *What is the name of the shoe with the red stripe?*

A: Adizero Boston 13

Reasoning

Q: *How much would it cost to buy the "Duramo 10" shoe in all available colorsways?*

A: 4 colors x $33 = $132



Grounding

Click on "Filter & Sort"

click (91.2, 12.9)

Favorite the "Questar 3"

click (94.9, 59.1)

Grounding

(Screenshot, description) --> click

Screenshot QA

(screenshot, question) --> answer

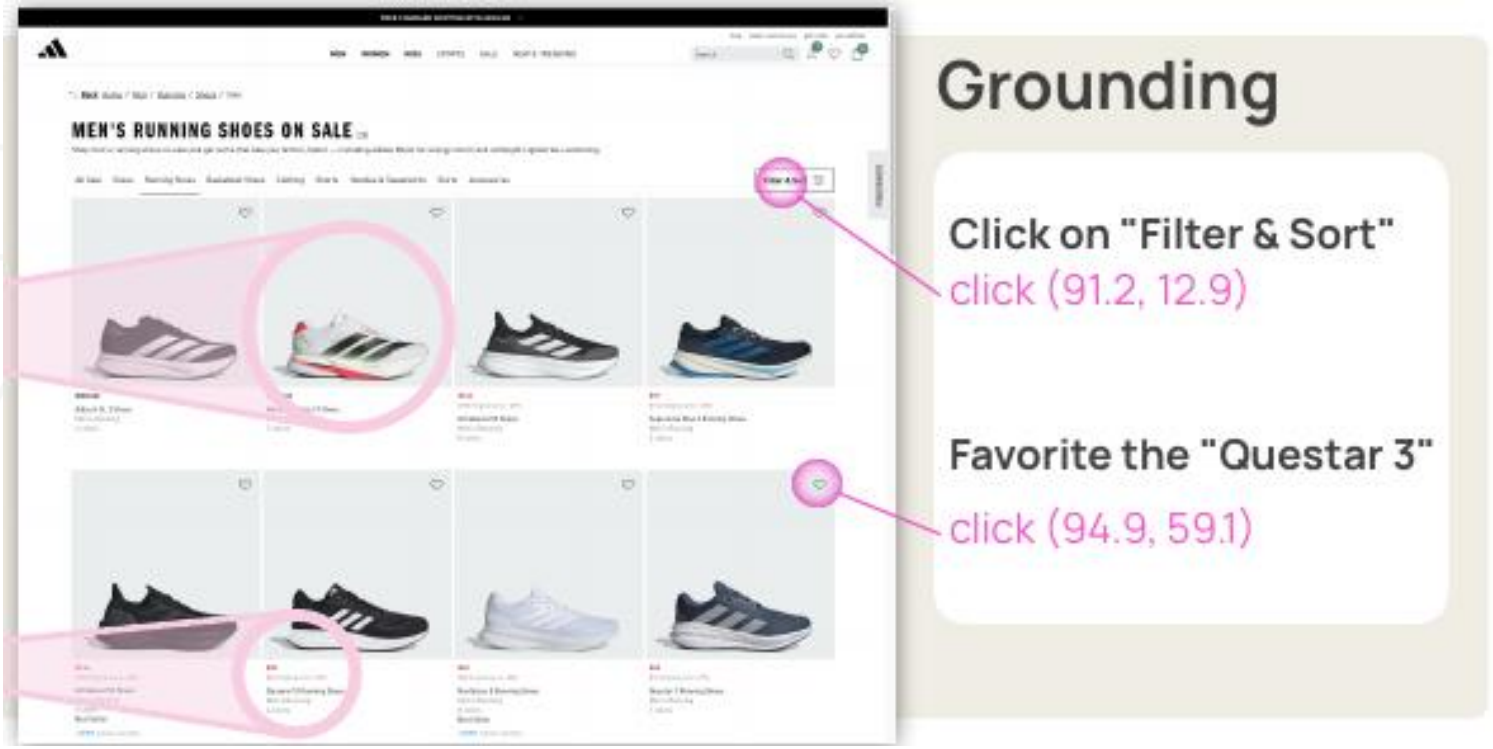
Grounding and Screenshot QA provide direct supervision for GUI perception.

Gupta et al., 2026, §2.3.

Georgia Tech

GUI perception: learning where to click

Complete tasks | Atomic skills | **GUI perception**



Task Format:

(Screenshot, description) --> click

Clickable element as Target

All clickable elements in the AxTree can be prediction target

Automatic Description

Corresponding AxTree names and roles are converted into descriptions

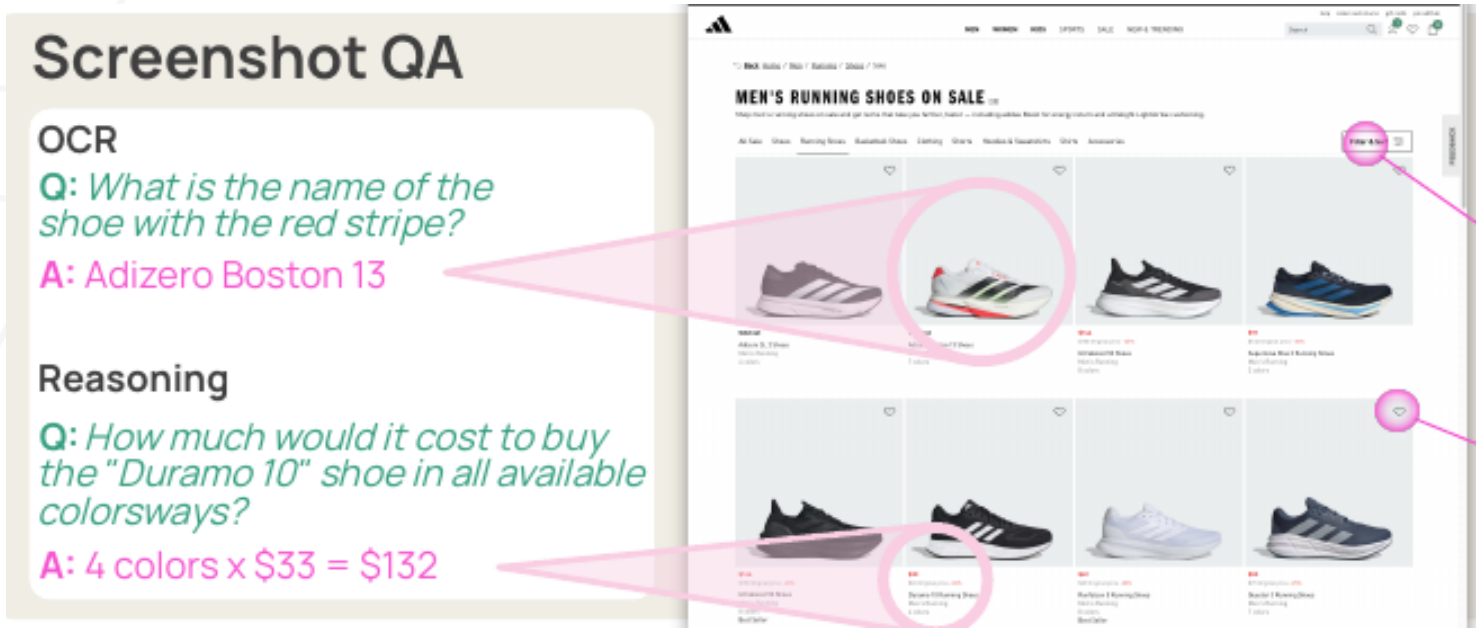
Click coordinate is randomly sampled within each element's bounding box.

Gupta et al., 2026, Fig. 2 and §2.3

Georgia Tech

GUI perception: reading the screenshot

Complete tasks | Atomic skills | **GUI perception**



OCR Question:

Extract information precisely

Affordance Question:

Identify available actions on the page

Summarization Question:

Understand the overall content or purpose of a page

AxTree-generated questions, filtered to remove AxTree-specific descriptions

Gupta et al., 2026, Fig. 2 and §2.3

Georgia Tech

Data and final training mixture

Dataset / Source	Trajectories	Steps	Domains	Avg steps	Mixture Ratio	Open-source
Mind2Web [6]	2.4K	17K	137	7.3	–	Yes
AutoWebWorld [21]	11.6K	255K	29	21.9	–	Yes
Fara [17]	145K	1.01M	–	6.9	–	No
MolmoWebMix-Traj	278.5K	2.2M	2.6K	13.2	0.80	Yes
Synthetic						
AxTree Single-Agent	70K	793K	1.3K	11.4	0.35	Yes
Axtree Multi-Agent	35K	438K	1.1K	12.5	0.18	Yes
Axtree Atomic Skills	5.5K	68.7K	540	12.4	0.02	Yes
Node-Traversal	16K	151K	833	9.5	0.02	Yes
Human						
Human	36K	623K	1.1K	20.8	0.18	Yes
Human Skills	116K	781K	1.1K	6.8	0.05	Yes
MolmoWebMix-Perception	–	10.5M	–	–	0.20	Yes
PixMoPoints + SyntheticGround	–	8.3M	–	–	0.15	Yes
ScreenshotQA	–	2.2M	–	–	0.05	Yes

Gupta et al., 2026, Table 2, §3.3 and §4.4.

Open data recipe
Large-Scale dataset

Makeup:
Task trajectories, atomic skills, and GUI perception

Human data: 23%

18% tasks + 5% skills

Georgia Tech

Training: a single SFT stage

SFT from: Molmo2 [4]

Qwen3 language model + SigLIP2 vision encoder

Single-Image Checkpoint

Training choice	Reported setup
Objective	One supervised fine-tuning stage
Updated components	Language model, vision encoder, adapter
Compute	64 H100s, global batch 128, up to 50K steps
Data exposure	≈3.2 epochs on the data mixture

Gupta et al., 2026, §3.3 and App. B.2

Georgia Tech

Evaluation: four live benchmarks

Benchmark	Task coverage	Judge used by MolmoWeb
WebVoyager	Common web workflows	GPT-4o
Online-Mind2Web	Realistic web tasks	o4-mini / WebJudge
DeepShop	Online shopping	GPT-4o
WebTailBench	Less common web workflows	WebVoyager judge †

Outcome: live task success

WebVoyager uses the Fara-filtered task set. † identifies a judge substitution.

Gupta et al., 2026, §4.1 and Table 4

Georgia Tech

Evaluation: browser environment and protocol

Browserbase Parallel live browser sessions with some CAPTCHA-solving support

Setting	Reported protocol
Task dates	Update outdated dates in time-sensitive instructions
Agent budget	Up to 100 steps per task
Repeated evaluation	3–5 runs per evaluation, average score
Environment errors	Up to 10 trajectory retries
Sampling	Top-p sampling, p = 0.8 and temperature = 0.7

Gupta et al., 2026, §4.1.

Results: live task success

Model	WebVoyager	Online-Mind2Web	DeepShop	WebTailBench
UI-TARS-1.5-7B	66.4	31.3	11.6	19.5
Fara-7B	73.5	34.1	26.2	38.4
MolmoWeb-4B	75.2	31.3	35.6	43.8 †
MolmoWeb-8B	**78.2**	**35.3**	**42.3**	**49.5 †**

What the table says beyond the ranking

- Gains over Fara-7B are uneven: +16 on DeepShop, +11 on WebTailBench, +5 on WebVoyager, +1 on Online-Mind2Web. The data recipe helps most on the harder, less common tasks.
- 4B to 8B adds 3–7 points everywhere, while 1% to 100% of the data adds 24 (next slide). Data moves results more than model size in this range.
- Online-Mind2Web is where open models fall furthest behind: no open 7–8B model clears 36, while closed agents reach 57–65.
- All numbers are judge-scored full-task success. No step-level metric is reported, so per-step errors are invisible; compare Mind2Web's 52% step vs. 5% task.

Gupta et al., 2026, Table 4 and §4.2. Task success (%), 100 steps. Baselines from Fara [17]. † WebVoyager judge, so the WebTailBench gap mixes two judges.

Georgia Tech

Ablation: how much data is needed?

Training-data fraction	WebVoyager	Online-Mind2Web
1%	44.5	11.7
10%	63.2	20.4
100%	**68.5**	**21.9**

10% of the data already reaches about 90% of full performance

- Diminishing returns: 1% to 10% adds 19 and 9 points; 10% to 100% adds only 5 and 1.5.
- Adding human data on top of synthetic barely moves the score: 67.8 to 68.5 on WebVoyager, 22.0 to 21.4 on Online-Mind2Web (Table 5b).
- Earlier data mixture and a 30-step budget, which is why 100% here (68.5) sits below the main-table 78.2.

Gupta et al., 2026, Table 5 and §4.4

Georgia Tech

Ablation: human vs. synthetic demonstrations

Trajectory source	Trajectories	DeepShop	WebVoyager	Online-Mind2Web
Human	2.7K	19.8	35.4	9.0
AxTree agent	**2.7K**	**24.4**	**53.0**	**16.8**

Agent demonstrations taught better on the same tasks

- Same 2.7K task instructions, both sets mixed with the same perception data.
- Agent trajectories are consistent and judge-filtered; human ones include detours and backtracking. Once synthetic trajectories are in the mix, the model almost always emits the generic scroll and stops using scroll_at, which the human-only model still produces.
- This is privileged supervision: the demonstrations come from a teacher that reads the page structure, and the student must reproduce them from pixels alone (next slide).
- One study at small scale. It says consistency matters, not that human data is useless.

Gupta et al., 2026, Table 6 and §4.4

Georgia Tech

Privileged supervision: teacher vs. student

Benchmark	AxTree teacher, 100 steps	MolmoWeb-8B	Gap
WebVoyager	85.6	78.2	7.4
Online-Mind2Web	44.8	35.3	9.5
DeepShop	55.3	42.3	13.0
WebTailBench †	63.5	49.5	14.0

Training-time input is richer than inference-time input

- The teacher (Gemini-3-Flash) sees only the accessibility tree and clicks by element ID. Screenshots are recorded for the student, but the teacher never sees them.
- Besides the observation space, the paper lists three contributors: Gemini is presumably much larger, the student must point in pixels instead of naming an element, and it must do OCR implicitly.
- The gap is largest on DeepShop and WebTailBench. Because the teacher is also larger, the gap is an upper bound on the cost of pixels over structure.
- Step budget is a confound too: the 30-step teacher (74.4 / 34.8) falls below the 100-step student on WebVoyager and Online-Mind2Web.

Gupta et al., 2026, §2.1, Table 4 and §4.2. † both scored with the WebVoyager judge.

Georgia Tech

What the ablations do and do not show

Established

- MolmoWeb starts from a pretrained Molmo2 checkpoint
- More data helps, with diminishing returns
- On matched tasks, synthetic trajectories teach better than human ones
- Human data adds little on top of synthetic

Still open

- How much of the 78.2 is new, vs. what Molmo2 could already do: no Molmo2 baseline is reported
- Only human vs. synthetic trajectories are ablated. Perception, skill, node-traversal and multi-agent data are never removed to measure their effect

A control we would run

Evaluate the untouched Molmo2 checkpoint on the benchmarks first, then add each data source in matched amounts with prompts, actions, step budget and judge held fixed. This is our proposal, not a result from the paper.

Gupta et al., 2026, §§3.3, 4.2, 4.4

Georgia Tech

Test-time scaling: pass@k

MolmoWeb-8B, 100 steps	pass@1	pass@2	pass@3	pass@4
WebVoyager	78.2	89.0	92.8	94.7
Online-Mind2Web	35.3	48.5	55.8	60.5

More attempts raise the ceiling, not single-run reliability

- pass@k: a task counts as solved if any of k independent rollouts succeeds, as judged by an LLM. Estimated from 5 rollouts per task.
- Parallel beats longer: three 30-step runs with judge selection reach 86.2% on WebVoyager, vs. 78.2% for one 100-step run.
- The paper's reading: parallel rollouts mitigate compounding error, where one wrong action derails a single rollout. This is error propagation, measured.
- Caveat: the same kind of judge picks the winner and scores it, so real-world best-of-N would likely be lower.

Gupta et al., 2026, Eq. 1, Fig. 6 and §4.3

Failure modes: thought vs. action

Thought	Emitted action	Observed outcome
“Type the query, then press Enter.”	keyboard_press(Enter)	Nothing was typed. No progress.

Perception

- OCR fails on small text
- Misreads long passages when answering questions

Action

- Rare actions are weak: scroll_at, drag-and-drop, hover
- With synthetic data, scrolls the page instead of the element

Decision

- Thought and action disagree (above)
- Stuck repeating one action; greedy decoding is worse (61.4 vs. 68.5 top-p)
- Vague or constraint-heavy instructions

Judge the executed action and the resulting page, not the explanation

This is the live-execution version of Mind2Web’s gap between step accuracy (52%) and task success (5%): one wrong small action breaks the whole task.

Gupta et al., 2026, §4.4, Table 7, §6; the example is from §6, outcome is schematic

Georgia Tech

Separating perception from decision errors

Model	ScreenSpot	ScreenSpot-v2
Qwen2.5-VL-7B	85.5	89.3
Fara-7B	86.7	89.3
Holo1-7B	87.4	89.9
MolmoWeb-4B (the agent)	87.2	89.5
MolmoWeb-Ground-8B (specialist)	88.7	91.8

Grounding is on par with peers, but single-click accuracy compounds

- These benchmarks score one click on one static screenshot. The agent matches Fara and Holo1; Ground-8B is a separate model trained only on grounding data.
- A ~10% miss rate per click is not small. Training trajectories average 13 steps, and $0.9^{13} \approx 0.25$: unrecovered misses alone would sink most tasks. Recovery is doing real work.
- The paper reports no counts per failure type and only end-of-task success, so it cannot say how much of the gap is perception vs. decision. A step-level breakdown is the analysis we would want.

Gupta et al., 2026, Table 8, Table 2 (avg. steps), §6

What persisted, and what changed?

	Mind2Web / MindAct	MolmoWeb
Goal	Follow web instructions	Follow web instructions
Page input	Textual HTML / DOM	Screenshot + URL / title
Action target	Element + operation / value	Coordinates + browser controls
Training signal	~1K human demonstrations	Synthetic + human + perception
Evaluation	Offline next-action prediction	Live task execution

Same formulation: instruction + current state + history → next action

What changed is what the model sees, how it acts, what it learns from, and how it is tested. The two papers are not evaluated on the same benchmark, so this is not an accuracy comparison.

Deng et al., 2023, §§2–4; Gupta et al., 2026, §§2–4

Limitations and implications

Shared limitations

- Reading: OCR fails on small text and long passages
- Instructions: ambiguous or constraint-heavy tasks degrade performance
- Recovery: a wrong click or a repeated scroll can derail a long task
- Coverage: Mind2Web is English-language, mostly U.S. sites; MolmoWeb works best when the target site is named
- Evaluation: LLM judges and changing websites make scores hard to reproduce

Deployment

- An incorrect answer becomes an incorrect action once the agent can act
- Mind2Web flags financial transactions and user control, and records state-changing clicks as Click (Fake) so they never execute
- MolmoWeb has no safety section; its live evaluation uses some CAPTCHA-solving, a capability Mind2Web lists as a risk
- Privacy: screenshots can contain personal data

Deng et al., 2023, §6 and App. B; Gupta et al., 2026, §4.1 and §6

Takeaways

Mind2Web

Real tasks and recorded human demonstrations made generalization measurable. Steps are learnable; whole tasks mostly were not.

MolmoWeb

A screenshot-only policy becomes practical through a broad, fully open data recipe that is mostly synthetic (57% of the mix vs. 23% human). The contribution is the data, not a new architecture.

Across both

Know the input, the supervision and the judge before interpreting a score. The goal has not changed: turn a user’s intent into reliable browser actions.

Synthesis of the two papers; mixture ratios from Gupta et al., 2026, Table 2

Discussion

01 Why did cleaner synthetic demonstrations beat human ones?

Same 2.7K tasks: agent demos 53.0 vs. human 35.4 on WebVoyager, and the AxTree teacher still beats the pixel student by 7–14 points. Is human data the wrong signal here, or just noisier? What does a student inherit when it only learns from a stronger model?

02 Which number would you trust before deploying this?

Mind2Web: 52% step success, 5% task success. MolmoWeb: 78% pass@1, 95% pass@4, LLM-judged, no Molmo2 baseline. Which one predicts tomorrow's behavior on an unseen site?

03 Do we still need agents that browse?

MolmoWeb argues API-based tools lack the generality of operating the browser as a human would. If sites start exposing agent endpoints, is a screenshot agent just a fallback? Or does the long tail of sites with no API keep it necessary?

Presenter questions; Q3 cites Gupta et al., 2026, §5

Georgia Tech

References

Mind2Web

Deng, X. et al. Mind2Web: Towards a Generalist Agent for the Web. NeurIPS 2023.

arxiv.org/abs/2306.06070

osu-nlp-group.github.io/Mind2Web

MolmoWeb

Gupta, T. et al. MolmoWeb: Open Visual Web Agent and Open Data for the Open Web. arXiv 2026.

arxiv.org/abs/2604.08516

github.com/allenai/molmoweb

Also cited

Xue, T. et al. An Illusion of Progress? Assessing the Current State of Web Agents (Online-Mind2Web). 2025.

He, H. et al. WebVoyager. ACL 2024. · Yao, S. et al. WebShop. NeurIPS 2022. · Zhou, S. et al. WebArena. ICLR 2024.

Figures and tables are credited on their slides; simplified tables are redrawn from the papers.

Georgia Tech