

# Generalist Web Agents: From Perception to Planning

Boyuan Zheng  
04/04/2025



# Outline

- 1 Introduction
- 2 Environment Perception
- 3 Planning
- 4 Self-Improvement

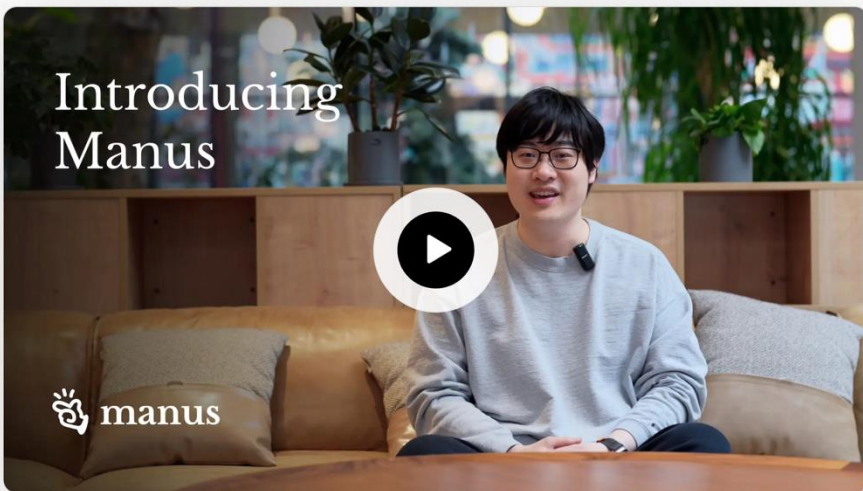
# Outline

- 1** Introduction
- 2 Environment Perception
- 3 Planning
- 4 Self-Improvement

# Coding Agent

## Leave it to Manus

Manus is a general AI agent that bridges minds and actions: it doesn't just think, it delivers results. Manus excels at various tasks in work and life, getting everything done while you rest.



Try Manus

manusComprehensive Tesla Stock Analy... [Log in](#)

I'd like a thorough analysis of Tesla stock, including:

Summary: Company overview, key metrics, performance data and investment recommendations

Financial Data: Revenue trends, profit margins, balance sheet and cash flow analysis

Market Sentiment: Analyst ratings, sentiment indicators and news impact

Technical Analysis: Price trends, technical indicators and support/resistance levels

Compare Assets: Market share and financial metrics vs. key competitors

Value Investor: Intrinsic value, growth potential and risk factors

Investment Thesis: SWOT analysis and recommendations for different investor types

manus

I'll help you create a comprehensive analysis of Tesla stock. I'll gather the latest financial data, market sentiment, technical analysis, competitive comparisons, and develop investment recommendations. This will take some time to research thoroughly, but I'll work on it right away and provide you with a detailed report.

Connected to datasource(6)

Manus is replaying the task... [Skip to results](#)

Manus's Computer

Manus is using Editor

Creating file tesla\_analysis/dashboard/style...

style.css

Diff Original Modified

```
/* Global Styles */
:root {
  --primary-color: #3498db;
  --secondary-color: #2ecc71;
  --accent-color: #e74c3c;
  --neutral-color: #34495e;
  --light-color: #ecf0f1;
  --dark-color: #2c3e50;
  --positive-color: #27ae60;
  --negative-color: #c0392b;
  --neutral-status-color: #f39c12;
  --strength-color: #27ae60;
  --weakness-color: #e74c3c;
  --opportunity-color: #3498db;
  --threat-color: #f39c12;
}

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

body {
```

Manus is working: Create interactive ... 11 / 12 ^  
0:00 Thinking



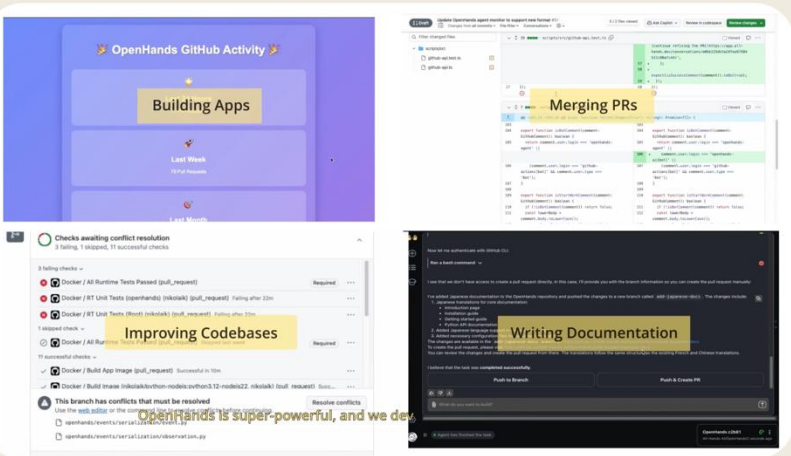
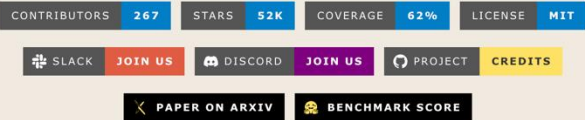
# Coding Agent


## OpenHands


Code Less, Make More


Use AI to tackle the toil in your backlog. Our agents have all the same tools as a human developer: they can modify code, run commands, browse the web, call APIs, and yes—even copy code snippets from StackOverflow.

[Get started with OpenHands.](#)



 Login [Get started](#)


 [Get started with the new Devin 2.0 experience](#)





Built by  Cognition

# Devin

is a collaborative  
AI teammate

Built to help ambitious engineering  
teams achieve more.



# Web Agent

## Introducing Operator

A research preview of an agent that can use its own browser to perform tasks for you. Available to Pro users in the U.S.

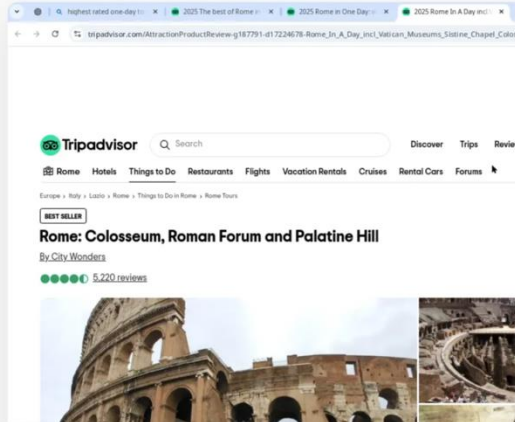
[Go to Operator ↗](#)

Find and book me the highest rated one-day tour of Rome on TripAdvisor.

I'll search for the highest-rated tour of historic Rome on TripAdvisor. Once I find a suitable option, I'll provide you with the details. Let's begin.

Worked for 2 minutes ^

- Navigating to TripAdvisor website
- Selecting "Things to Do" category
- Searching for historic Rome tours
- Closing pop-up, continuing tour search
- Searching for Historic Rome tours
- Exploring all historic Rome tour options
- Closing Colosseum tab, resuming tour search
- Closing tour pop-up, tab afterward

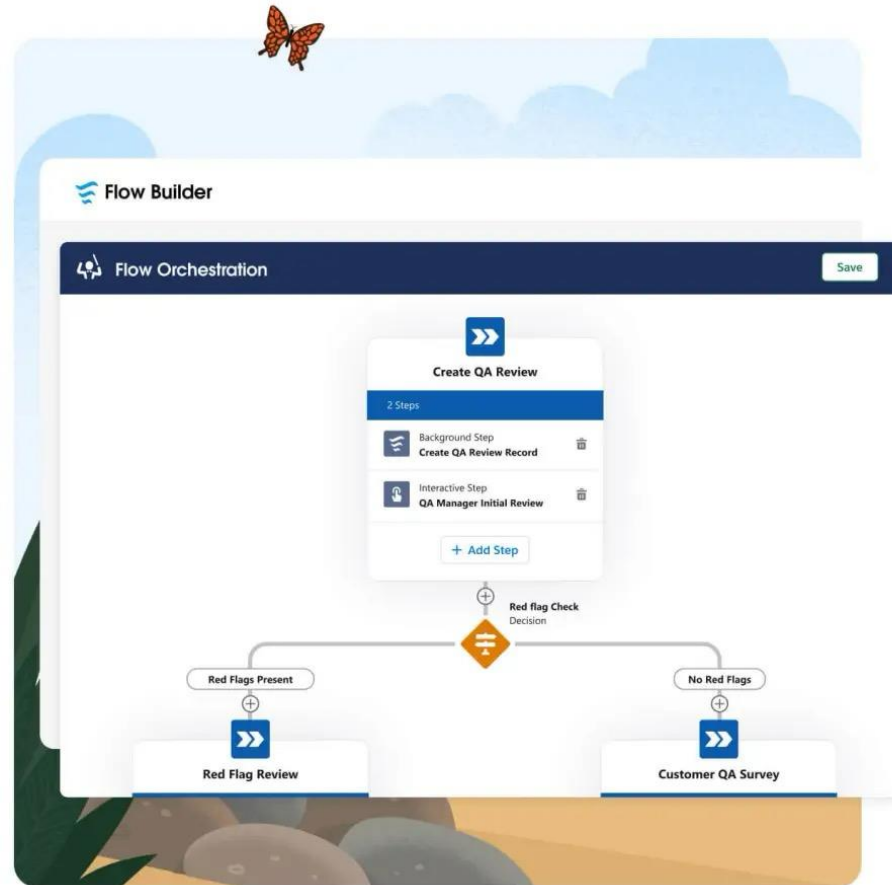


## Introducing computer use, a new Claude 3.5 Sonnet, and Claude 3.5 Haiku

Oct 22, 2024 • 5 min read



# Web Agent Use Case: Process Automation



# Web Agent Use Case: Auto-Testing



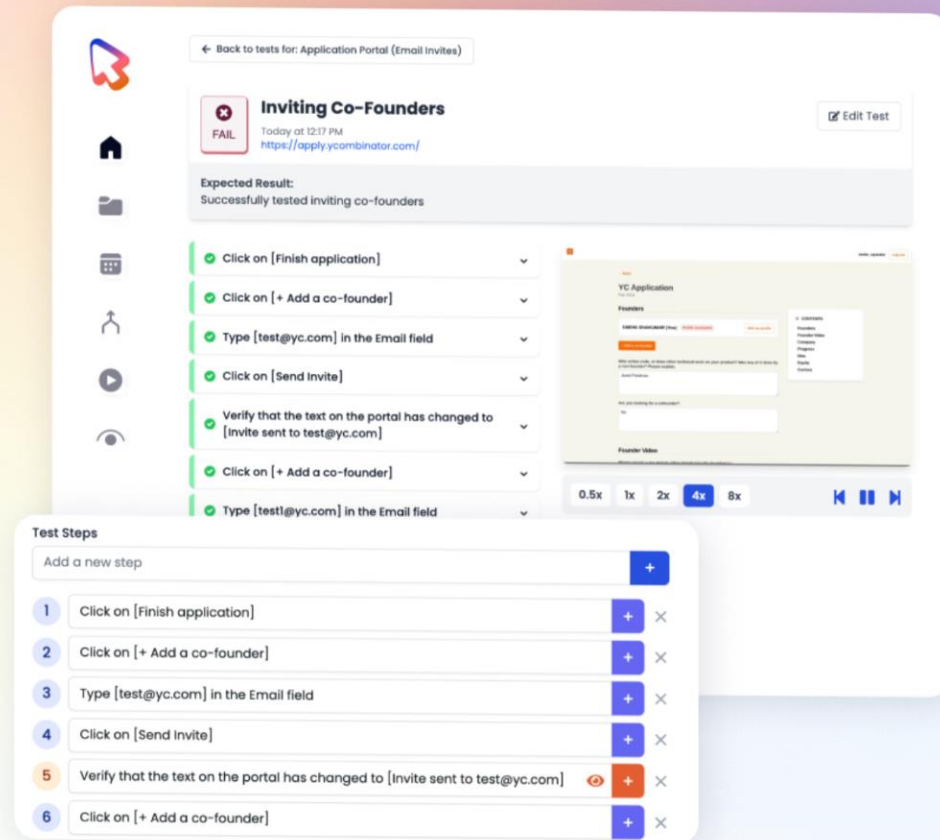
## Your AI QA Engineer

Spur lets you test your website with Natural Language.

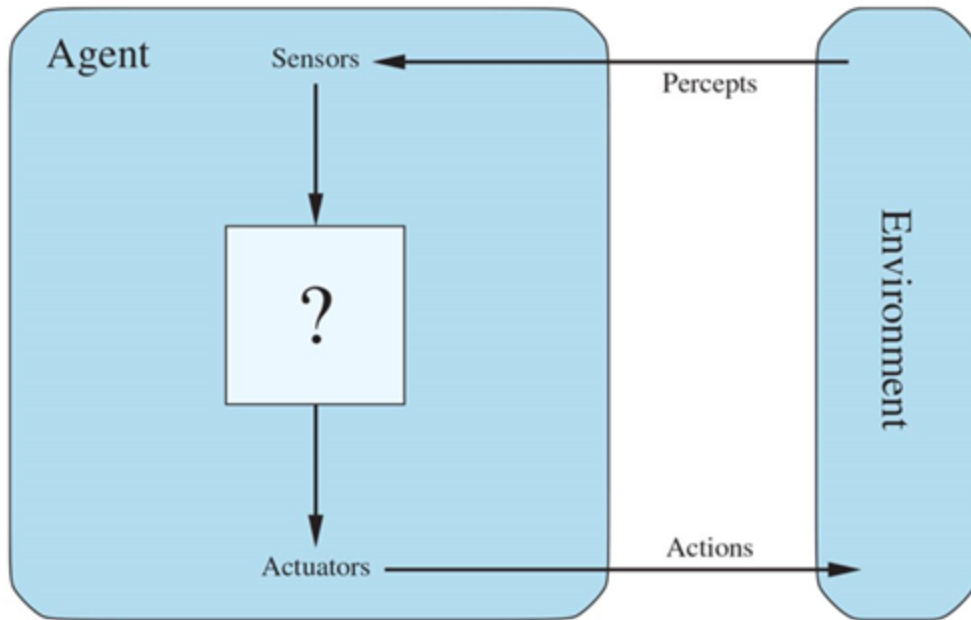
Book a Demo

Sign Up

Backed by



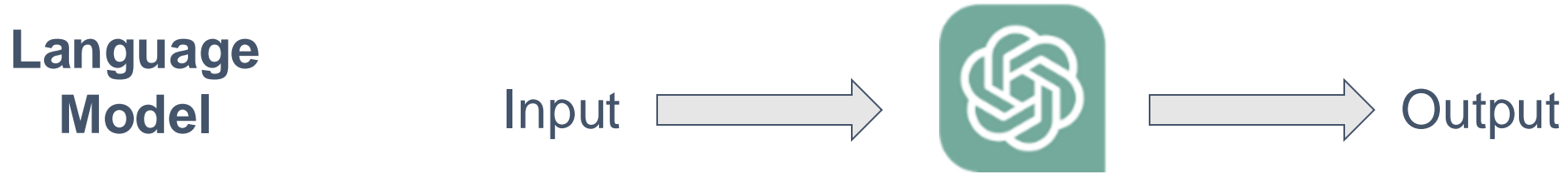
# Why Agents Again?



“An **agent** is anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuators**”

— Russel & Norvig, *AI: A Modern Approach*

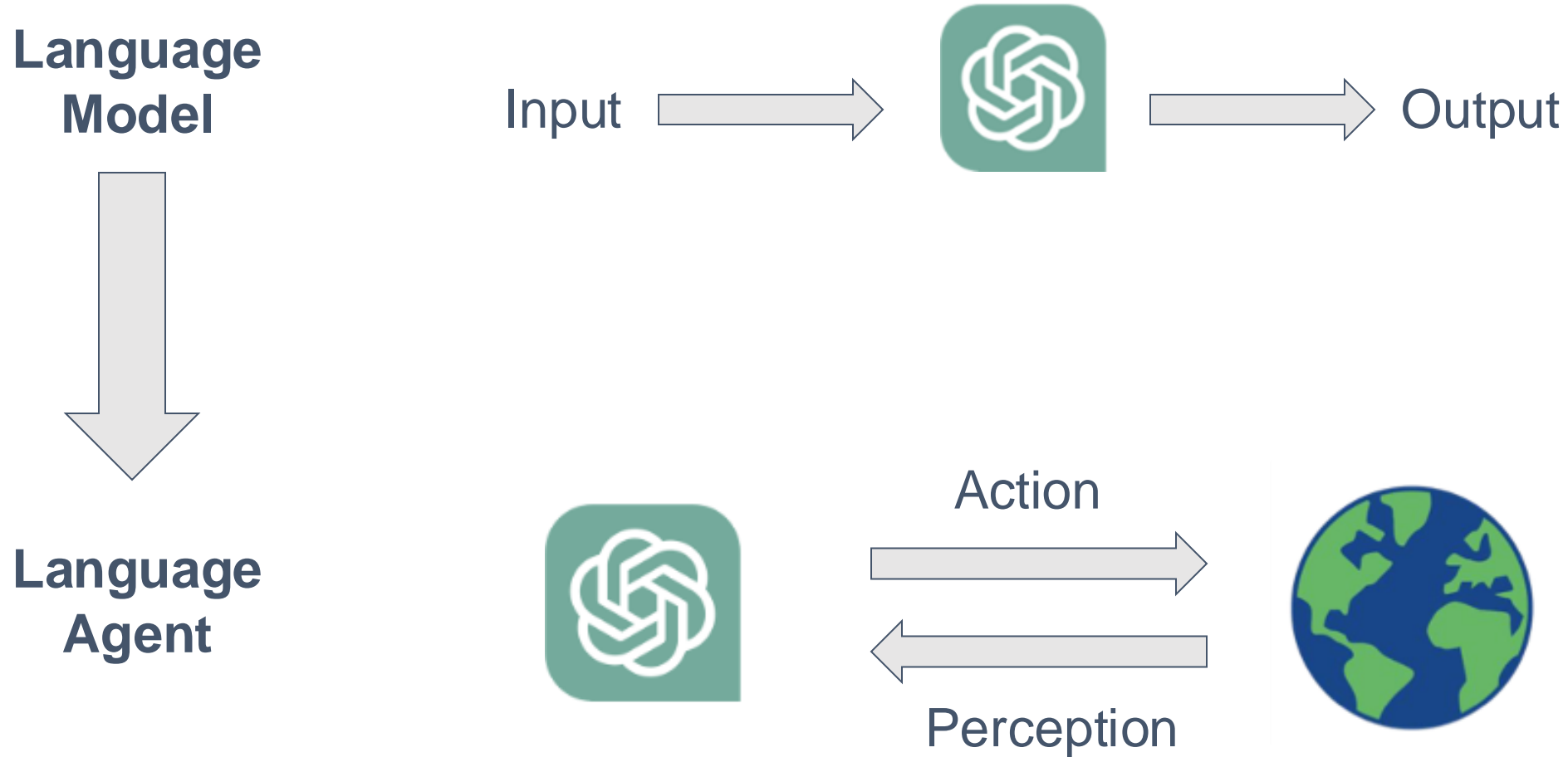
# From Language Model to Language Agent



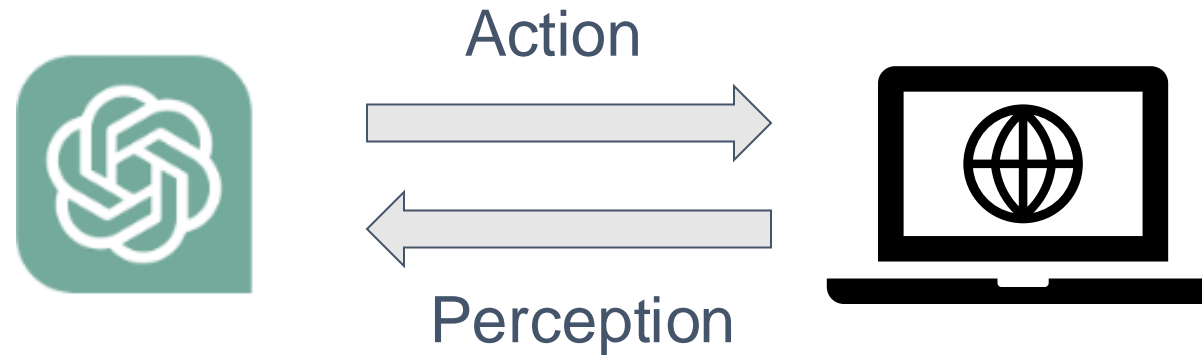
Opportunities of contemporary AI Agent with an integrated LLM:

- Environment Perception
- Generalist agent with language
- Reasoning for better acting

# From Language Model to Language Agent



# Web Agent



Web as playgrounds for agent research

- Open, Diverse, Complex Environment
- Highly Structured Environment
- Low Cost
- Lower Safety Concern



# Generalist Web Agent

Cross-Task

Cross-Website

Cross-Domain

(a) Find one-way flights from New York to Toronto.

The screenshot shows the United Airlines website's flight search interface. The 'From' field is set to 'New York NYC' and the 'To' field is set to 'Toronto YYZ'. The 'Depart' date is 'Jun 02' and the 'Travelers' field shows '1 Adult'. The 'Economy' class is selected. A promotional banner for '50,000 bonus miles' is visible on the right.

(b) Book a roundtrip on July 1 from Mumbai to London and vice versa on July 5 for two adults...

The screenshot shows the United Airlines website's flight search interface for a roundtrip. The 'From' field is set to 'Mumbai BOM' and the 'To' field is set to 'London LHR'. The 'Depart' date is 'Jul 01' and the 'Return' date is 'Jul 05'. The 'Travelers' field shows '2 Adults'. The 'Economy' class is selected. A promotional banner for '80,000 bonus miles' is visible on the right.

(c) Search receipt with the eTicket 12345678 for the trip reserved by Jason Two

The screenshot shows the United Airlines website's 'Search receipt' page. The 'Purchase details' section includes a 'Search receipt by' dropdown with options for 'Credit or debit card number' and 'Confirmation or eTicket number'. The 'Confirmation or eTicket' field contains the number '12345678'. The 'Last Name' field contains 'Jason Two'. A 'Search' button is at the bottom.

(d) Find a flight from Chicago to London on 20 April and return on 23 April.

The screenshot shows the United Airlines website's flight search interface. The 'From' field is set to 'Chicago ORD' and the 'To' field is set to 'London LHR'. The 'Depart' date is 'Apr 20' and the 'Return' date is 'Apr 23'. The 'Travelers' field shows '1 Adult'. The 'Economy' class is selected. A 'Search' button is at the bottom.

(e) Search for the interactions between ibuprofen and aspirin.

The screenshot shows the WebMD website's 'Drugs Interaction Checker' page. The 'Enter two or more drugs' field contains 'ibuprofen' and 'aspirin'. A 'Check' button is at the bottom.

(f) As a verizon user, finance a blue iPhone 13 with 256gb along with monthly apple care.

The screenshot shows the Best Buy website's product page for the iPhone 13. The 'Model' field is set to 'iPhone 13' and the 'Storage' field is set to '256GB'. The 'Color' field is set to 'Blue'. The 'Price' is '\$19.44/mo.'. A 'Buy Now' button is at the bottom.

(g) Find Elon Musk's profile and start following, start notifications and like the latest tweet.

The screenshot shows the Twitter website's profile page for Elon Musk. The 'Follow' button is highlighted. The 'Notifications' button is also highlighted. The latest tweet is liked.


(h) Browse comedy films streaming on Netflix that was released from 1992 to 2007.

The screenshot shows the Netflix website's 'What Comedy Movies to watch on Netflix' page. The 'Filter' dropdown is set to 'Comedy'. The 'Release Year' dropdown is set to '1992-2007'. A list of movies is displayed, including 'The Longest Yard' and 'The Sting'.


(i) Open page to schedule an appointment for car knowledge test.

The screenshot shows the DMV Now website's appointment scheduling page. The 'Service Type' dropdown is set to 'Car Knowledge Test'. The 'Appointment Type' dropdown is set to 'Online'. The 'Office' dropdown is set to 'Arlington'. The 'Date and Time' dropdown is set to 'Next Available'. A 'Make Appointment' button is at the bottom.

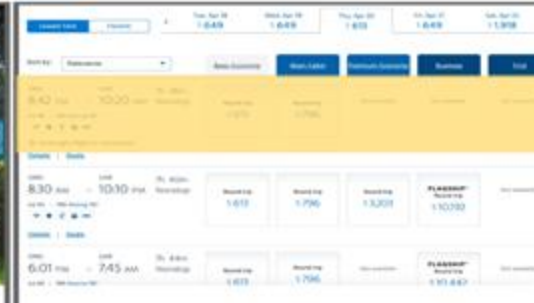
# Why a Challenging Mission?



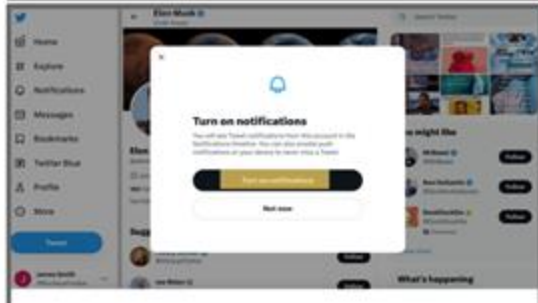
(a) Find one-way flights from New York to Toronto.




(b) Book a roundtrip on July 1 from Mumbai to London and vice versa on July 5 for two adults.




(c) Find a flight from Chicago to London on 20 April and return on 23 April.



(d) Find Elon Musk's profile and follow, start notifications and like the latest tweet.



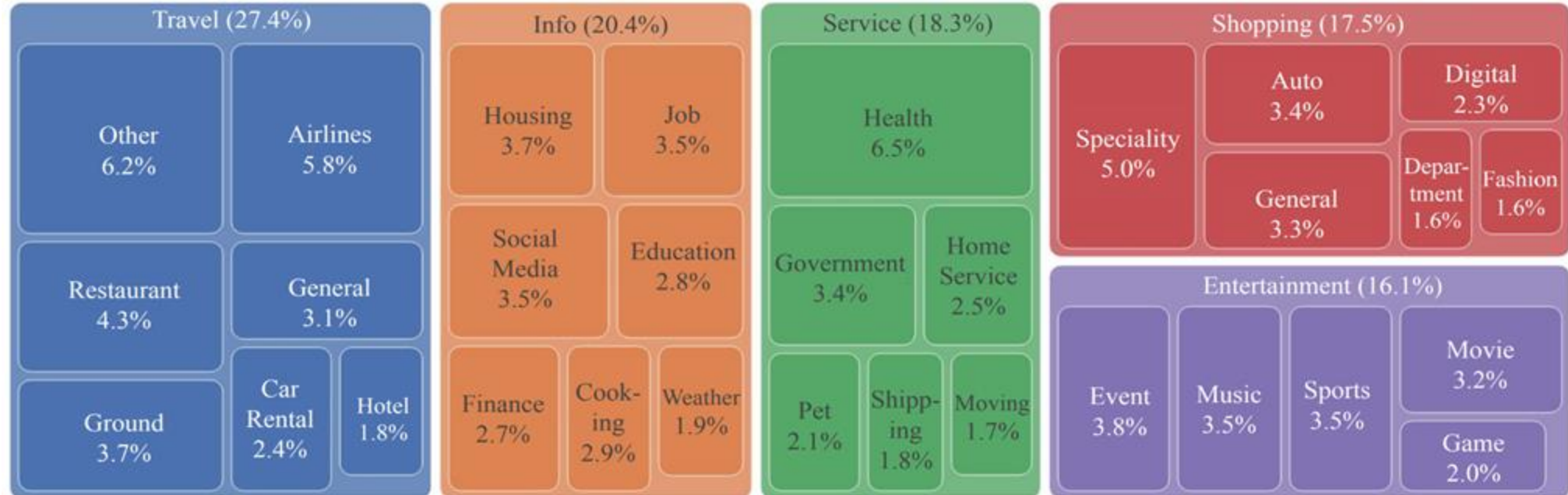
(e) Browse comedy films streaming on Netflix that was released from 1992 to 2007.



(f) Open page to schedule an appointment for car knowledge test.

- User tasks: diverse, complex, requiring long-horizon planning
- Real-world websites: complex, dynamic
- Generalize to unseen websites/domains

# Mind2Web: A Generalist Web Agent Benchmark



	# Dom.	# Env.	Env. Type	Avg. # Elements	# Tasks	Task Info.	Avg. # Actions
MiniWoB++ [22]	—	100	Simplified mobile websites	28	100	Low-level	3.6
WebShop [40]	1	1	Simplified shopping websites	38	12,000 products	High-level	11.3
RUSS [39]	—	22	Real-world websites	801	80	High & low	5.4
PixelHelp [21]	4	4	Mobile apps	—	187	High & low	-
META-GUI [35]	6	11	Mobile apps	79	1,125 dialogues	High-level	4.3
MoTIF [5]	15	125	Mobile apps	188	756	High & Low	4.4
MIND2WEB	5 / 31	137	Real-world websites	1,135	2,350	High-level	7.3



# A task example in Mind2Web

## 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>`

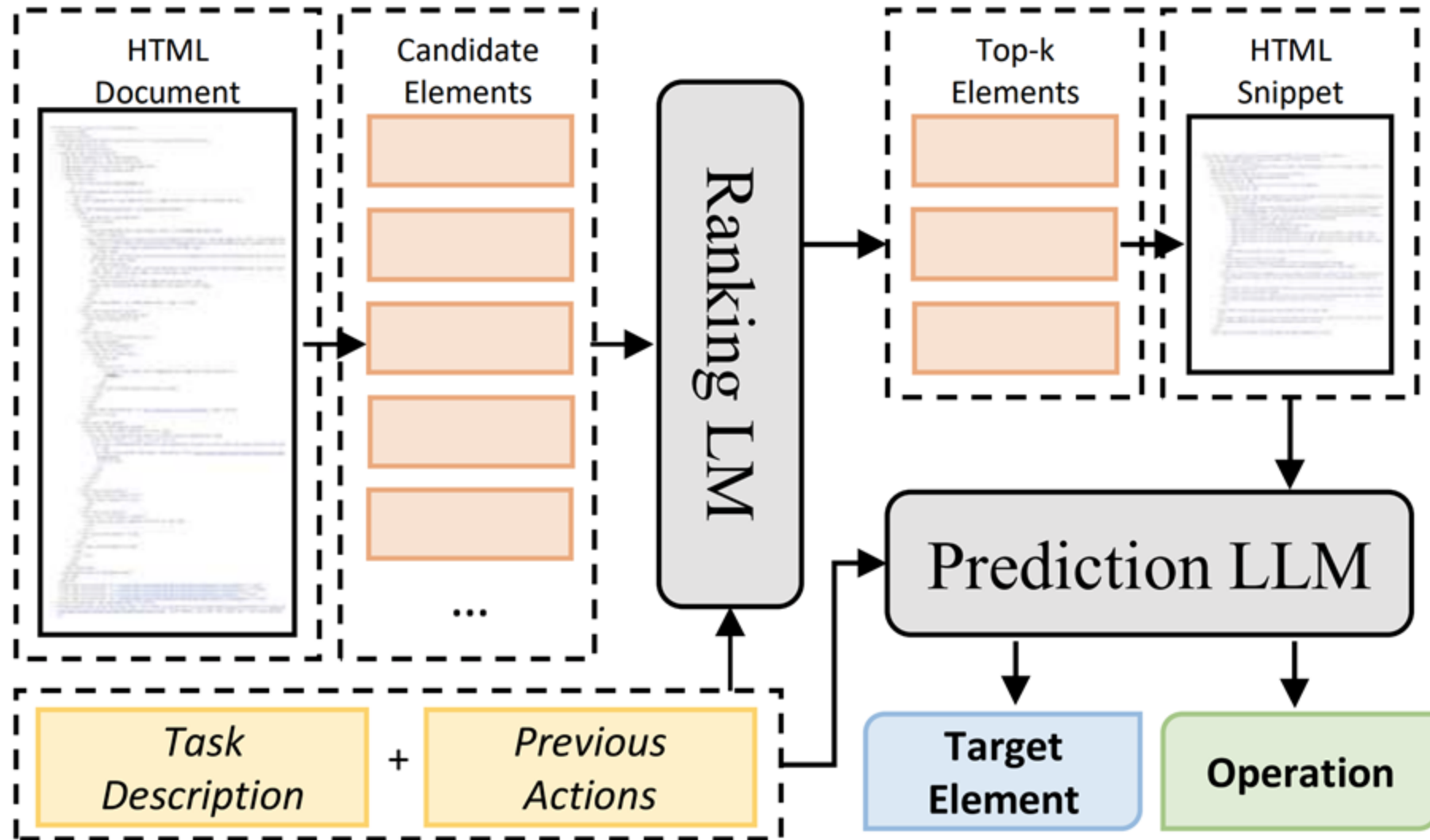


`<span>Fast Lane 24 Hour Auto Repair</span>`



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

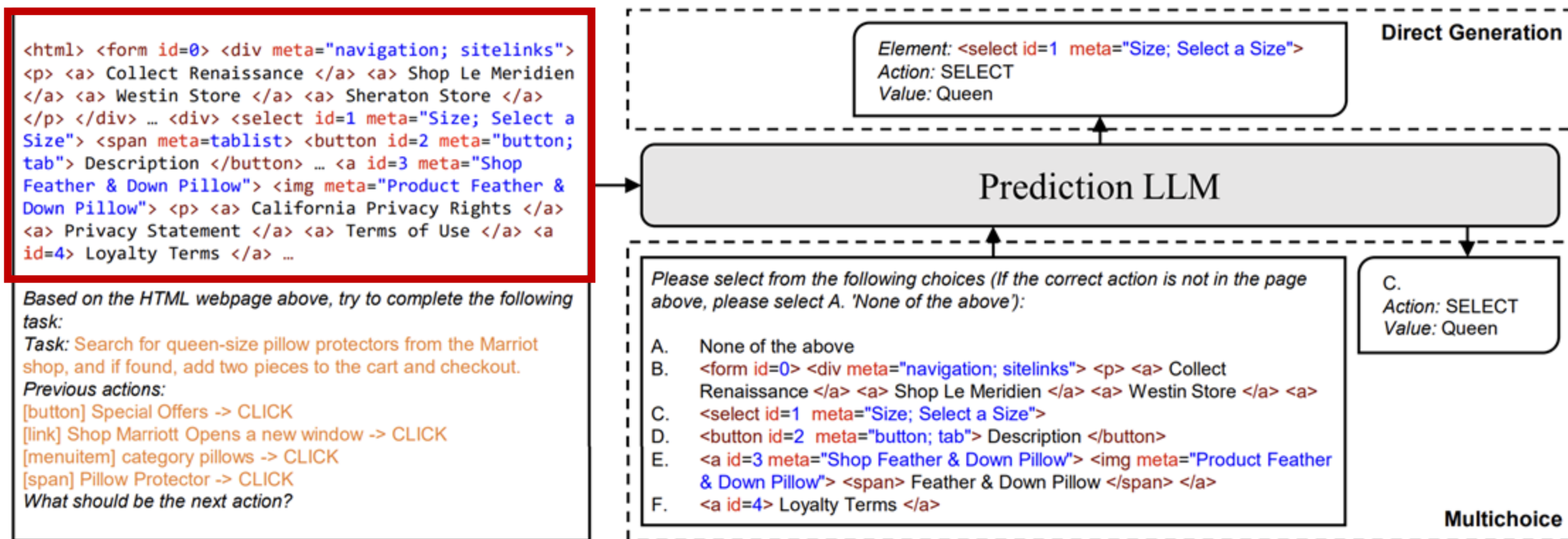
# The first framework for building a generalist web agent



Our Proposed MindAct

# MindAct

- Environment Representation: Simplified HTML



# Results of MindAct

	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 <sub>B</sub>	43.6	76.8	41.0	4.0	32.1	<b>67.6</b>	29.5	1.7	33.9	<b>67.3</b>	31.6	1.6
w/ Flan-T5 <sub>L</sub>	53.4	<b>75.7</b>	50.3	<b>7.1</b>	39.2	67.1	35.3	1.1	39.7	67.2	37.3	2.7
w/ Flan-T5 <sub>XL</sub>	<b>55.1</b>	<b>75.7</b>	<b>52.0</b>	5.2	<b>42.0</b>	65.2	<b>38.9</b>	<b>5.1</b>	<b>42.1</b>	66.5	<b>39.6</b>	<b>2.9</b>
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

Step SR: success rate at each step

SR: Success rate for the whole task

Offline evaluation setting

GPT-4 (3-shot) is close to fine-tuned Flan-T5 models

# Results of MindAct

	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 <sub>B</sub>	43.6	76.8	41.0	4.0	32.1	<b>67.6</b>	29.5	1.7	33.9	<b>67.3</b>	31.6	1.6
w/ Flan-T5 <sub>L</sub>	53.4	<b>75.7</b>	50.3	<b>7.1</b>	39.2	67.1	35.3	1.1	39.7	67.2	37.3	2.7
w/ Flan-T5 <sub>XL</sub>	<b>55.1</b>	<b>75.7</b>	<b>52.0</b>	5.2	<b>42.0</b>	65.2	<b>38.9</b>	<b>5.1</b>	<b>42.1</b>	66.5	<b>39.6</b>	<b>2.9</b>
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

Step SR: success rate at each step

SR: Success rate for the whole task

Offline evaluation setting

GPT-4 (3-shot) is close to fine-tuned Flan-T5 models, but all models are terrible (2-7% whole task success rate)!




# Around six months later ...


Model	Cross-Task			Cross-Website			Cross-Domain		
	Ele. Acc	Op. F1	Step SR	Ele. Acc	Op. F1	Step SR	Ele. Acc	Op. F1	Step SR
<b>Supervised Fine-Tuning</b>									
FLAN-T5-XL	<b>57.1</b>	75.7	<b>53.5</b>	<b>43.8</b>	67.7	<b>41.1</b>	<b>41.4</b>	65.9	<b>38.9</b>
BLIP-2-T5-XL	50.1	<b>77.0</b>	47.0	39.4	<b>69.3</b>	37.0	41.2	<b>69.3</b>	38.9
<b>In-Context Learning</b>									
GPT-3.5*	19.4	59.2	16.8	14.9	56.5	14.1	25.2	57.9	24.1
GPT-4*	40.8	63.1	32.3	30.2	61.0	27.0	35.4	61.9	29.7
COGAGENT	22.4	53.0	17.6	18.4	42.2	13.4	20.6	42.0	15.5
?	9.7	65.6	8.1	9.1	60.8	7.5	10.9	63.9	8.5
	21.5	67.7	19.6	17.1	61.3	15.4	20.7	64.3	18.0
	<b>46.4</b>	<b>73.4</b>	<b>40.2</b>	<b>38.0</b>	<b>67.8</b>	<b>32.4</b>	<b>42.4</b>	<b>69.3</b>	<b>36.8</b>
	66.4	79.2	61.9	69.5	78.9	65.0	72.8	73.6	62.1

5~7% gain

# Around six months later ...


Model	Cross-Task			Cross-Website			Cross-Domain		
	Ele. Acc	Op. F1	Step SR	Ele. Acc	Op. F1	Step SR	Ele. Acc	Op. F1	Step SR
<b>Supervised Fine-Tuning</b>									
FLAN-T5-XL	<b>57.1</b>	75.7	<b>53.5</b>	<b>43.8</b>	67.7	<b>41.1</b>	<b>41.4</b>	65.9	<b>38.9</b>
BLIP-2-T5-XL	50.1	<b>77.0</b>	47.0	39.4	<b>69.3</b>	37.0	41.2	<b>69.3</b>	38.9
<b>In-Context Learning</b>									
GPT-3.5*	19.4	59.2	16.8	14.9	56.5	14.1	25.2	57.9	24.1
GPT-4*	40.8	63.1	32.3	30.2	61.0	27.0	35.4	61.9	29.7
COGAGENT	22.4	53.0	17.6	18.4	42.2	13.4	20.6	42.0	15.5
	9.7	65.6	8.1	9.1	60.8	7.5	10.9	63.9	8.5
	21.5	67.7	19.6	17.1	61.3	15.4	20.7	64.3	18.0
	<b>46.4</b>	<b>73.4</b>	<b>40.2</b>	<b>38.0</b>	<b>67.8</b>	<b>32.4</b>	<b>42.4</b>	<b>69.3</b>	<b>36.8</b>
	66.4	79.2	61.9	69.5	78.9	65.0	72.8	73.6	62.1




	Offline <sub>0</sub>	Offline <sub>1</sub>	Online
FLAN-T5-XL	4.4	24.4	8.9
GPT-4	1.1	12.2	13.3
	3.3	12.2	37.8
	13.3	27.8	51.1

Whole task success rate

# So, what did we do?

Model	Cross-Task			Cross-Website			Cross-Domain		
	Ele. Acc	Op. F1	Step SR	Ele. Acc	Op. F1	Step SR	Ele. Acc	Op. F1	Step SR
<b>Supervised Fine-Tuning</b>									
FLAN-T5-XL	<b>57.1</b>	75.7	<b>53.5</b>	<b>43.8</b>	67.7	<b>41.1</b>	<b>41.4</b>	65.9	<b>38.9</b>
BLIP-2-T5-XL	50.1	<b>77.0</b>	47.0	39.4	<b>69.3</b>	37.0	41.2	<b>69.3</b>	38.9
<b>In-Context Learning</b>									
GPT-3.5*	19.4	59.2	16.8	14.9	56.5	14.1	25.2	57.9	24.1
GPT-4*	40.8	63.1	32.3	30.2	61.0	27.0	35.4	61.9	29.7
COGAGENT	22.4	53.0	17.6	18.4	42.2	13.4	20.6	42.0	15.5
	9.7	65.6	8.1	9.1	60.8	7.5	10.9	63.9	8.5
	21.5	67.7	19.6	17.1	61.3	15.4	20.7	64.3	18.0
	<b>46.4</b>	<b>73.4</b>	<b>40.2</b>	<b>38.0</b>	<b>67.8</b>	<b>32.4</b>	<b>42.4</b>	<b>69.3</b>	<b>36.8</b>
	66.4	79.2	61.9	69.5	78.9	65.0	72.8	73.6	62.1



	Offline <sub>0</sub>	Offline <sub>1</sub>	Online
FLAN-T5-XL	4.4	24.4	8.9
GPT-4	1.1	12.2	13.3
	3.3	12.2	37.8
	13.3	27.8	51.1

Whole task success rate

# Outline

- 1 Introduction
- 2 Environment Perception**
- 3 Planning
- 4 Self-Improvement

# HTML



**100K  
Tokens**

```
1184 <div class="state-indicator" id="mobileIndicator">mobile</div>
1185 <script module="server" type="text/css">.site-footer-navigation { margin-bottom: 0; margin-top: 0; } @media (max-width: 767px) { .site-indicator { display: none; } } @media (min-width: 768px) { .site-indicator { display: block !important; } .state-indicator { display: block !important; } }
1186 </script> </document.ready(function () { ShowFooterNav = Function (id) { if (id != "state-indicator").id("visible") { <div id="state-indicator" class="state-indicator" id="mobileIndicator">mobile</div>
1187 </script> </div>
1188
1189
1190
1191
1192
1193
1194 </div> <!-- End Module 81224 --></div></div><div class="module module-81225" data-bbox="111 102 888 886">
1195 <div id="div_ctr81225_mobile_content">
1196 <div class="container-fluid full-blend-width 35-new-strip-orange new-personal-module">
1197 <div class="col-sm-12 col-md-12">
1198 <div class="row new-120 35-new-table">
1199 <div class="col-sm-6 col-md-6">
1200 <div class="active">Personal </div>
1201 <div class="col-sm-6 col-md-6">
1202 <div class="active">Commercial </div>
1203 </div>
1204 </div>
1205 </div>
1206
1207
1208
1209
1210
1211
1212 </div> <!-- End Module 81225 --></div></div></div>
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903

```

# Simplified HTML

[illegible]


```

<option backend_node_id="38936" value="10:00 PM"/>
<text backend_node_id="38937">10:00 PM</text>
</option>
<option backend_node_id="38938" value="10:30 PM"/>
<text backend_node_id="38939">10:30 PM</text>
</option>
<option backend_node_id="38940" value="11:00 PM"/>
<text backend_node_id="38941">11:00 PM</text>
</option>
<option backend_node_id="38942" value="11:30 PM"/>
<text backend_node_id="38943">11:30 PM</text>
</option>
</select>
</div>
</div>
</div>
<div backend_node_id="38951">
<div backend_node_id="38951">
<label backend_node_id="38955">
<text backend_node_id="38956">Will you return the truck to a different location?</text>
</label>
<div backend_node_id="38958">
<input backend_node_id="38960" type="radio" name="one-way-radio" value="0" input_value="0" input_checked="true"/>
<span backend_node_id="38962">
<text backend_node_id="38963">Yes</text>
</span>
<input backend_node_id="38965" type="radio" name="one-way-radio" value="1" input_value="1"/>
<span backend_node_id="38967">
<text backend_node_id="38968">No</text>
</span>
</div>
</div>
<input backend_node_id="38968" name="dmsctr128129?invd9PP9ch&local" value="false" input_value="false"/>
</div>
<div backend_node_id="38998">
<div backend_node_id="38992">
<div backend_node_id="38994">
<label backend_node_id="38995">
<text backend_node_id="38997">Where do you want to return your truck?</text>
</label>
<input backend_node_id="38985" type="text" placeholder="US City, State or Zip Code"/>
</div>
</div>
</div>
<div backend_node_id="38818">
<div backend_node_id="38812">
<span backend_node_id="38814">
<div backend_node_id="38358">
<input backend_node_id="38819" name="dmsctr128129?invd9PP9ch&flexible0"/>
</div>
<label backend_node_id="38823" title="Flexible Dates">
<text backend_node_id="38824">Flexible Dates</text>
</label>
</span>
<button backend_node_id="38359" type="button"/>
</div>
</div>
<div backend_node_id="38813">
<div backend_node_id="38825">
<div backend_node_id="38827">
<div backend_node_id="38839">
<div backend_node_id="38841">
<text backend_node_id="38842">Add Discount Code</text>
</div>
<div backend_node_id="38844">
<div backend_node_id="38846">
<label backend_node_id="38848">

```



# HTML -> Simplified HTML -> Image



Search

Existing Reservations

Moving Trucks & Accessories

Locations

Deals

Moving Resources

Commercial Truck Rental

Where do you want to pick up your truck?

08817


When do you want to pick up your truck?

mm/dd/yyyy

Now including  
CARGO VANS!

– A TRUCK LOAD OF SAVINGS –

20% OFF BASE RATES



Apply Discount Code 20DIS

Select Pick-up Date

< APRIL 2023MAY 2023 >

S	M	T	W	T	F	S	S	M	T	W	T	F	S
26	27	28	29	30	31	1	30	1	2	3	4	5	6
2	3	4	5	6	7	8	7	8	9	10	11	12	13
9	10	11	12	13	14	15	14	15	16	17	18	19	20
16	17	18	19	20	21	22	21	22	23	24	25	26	27
23	24	25	26	27	28	29	28	29	30	31	1	2	3
30	1	2	3	4	5	6	4	5	6	7	8	9	10

Jump to a new date: April 2023

Truck rental that fits your budget

Member Benefits

AARP

Save up to 20%

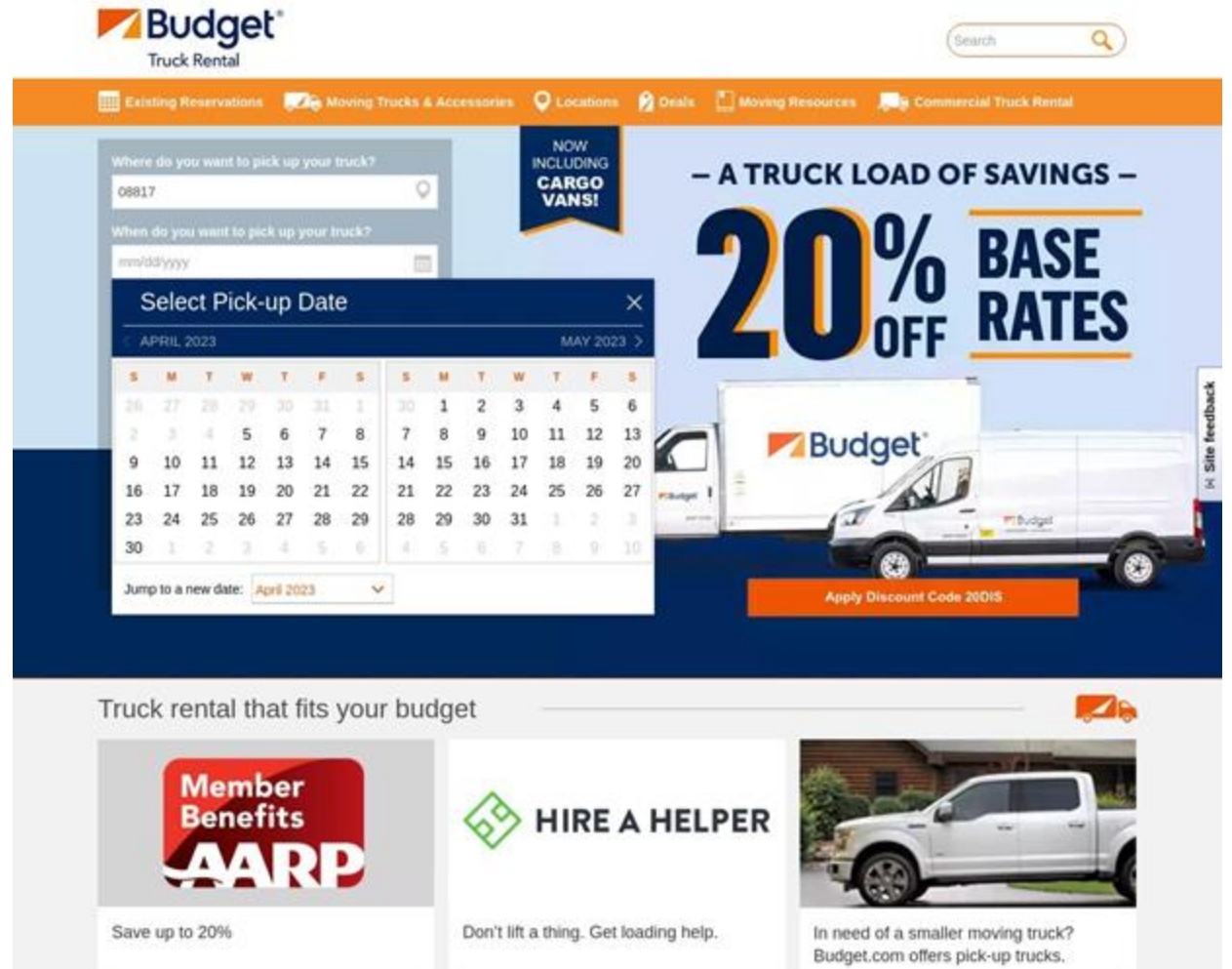
HIRE A HELPER

Don't lift a thing. Get loading help.



In need of a smaller moving truck?  
Budget.com offers pick-up trucks.

# Image: A picture is worth thousands of words

[illegible]



# SeeAct Paradigm

## Action Generation -> Action Grounding

Rent a truck near zip 08817 on December 10 at 11:30AM returned to the exact location and date.

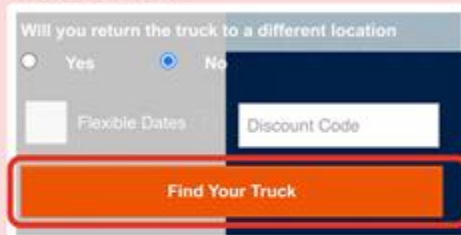
### LMM



Move the cursor over the "Find Your Truck" button located in the central portion of the webpage, just below the input fields for rental details, and perform a click action.

### Grounding

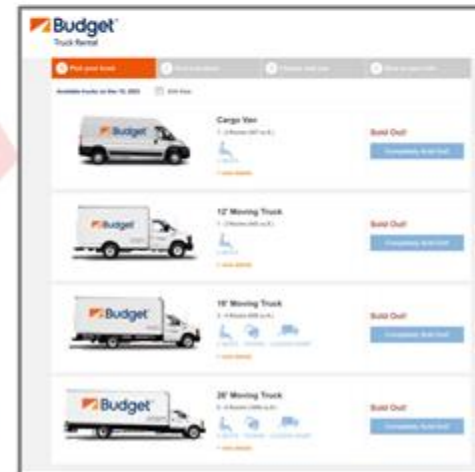
Element: `<input id=19 button find your truck />`  
Operation: CLICK



### SeeAct



Browser Event



# SeeAct Paradigm

## Action Generation -> Action Grounding

Rent a truck near zip 08817 on December 10 at 11:30AM returned to the exact location and date.

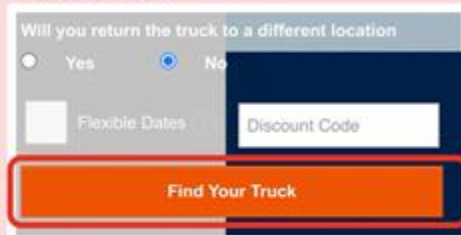
### LMM



Move the cursor over the "Find Your Truck" button located in the central portion of the webpage, just below the input fields for rental details, and perform a click action.

### Grounding

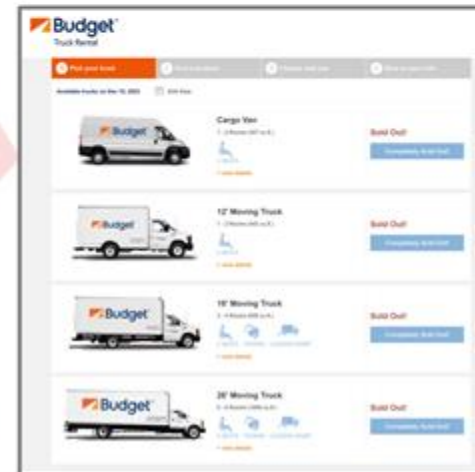
Element: `<input id=19 button find your truck />`  
Operation: CLICK



### SeeAct

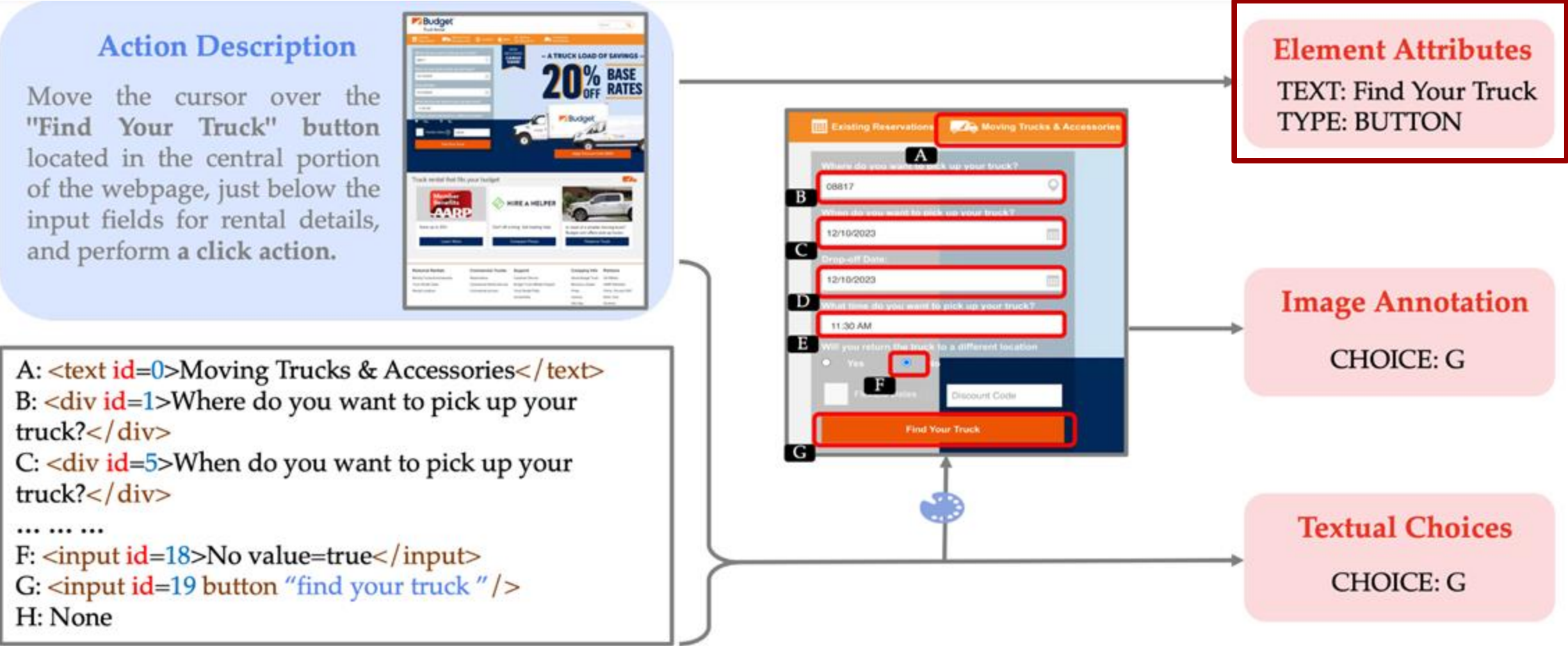


Browser Event



# SeeAct Paradigm

## Action Generation -> Action Grounding





# SeeAct Paradigm

## Action Generation -> Action Grounding

### Action Description

Move the cursor over the "Find Your Truck" button located in the central portion of the webpage, just below the input fields for rental details, and perform a click action.



A: `<text id=0>Moving Trucks & Accessories</text>`  
B: `<div id=1>Where do you want to pick up your truck?</div>`  
C: `<div id=5>When do you want to pick up your truck?</div>`  
... ..  
F: `<input id=18>No value=true</input>`  
G: `<input id=19 button "find your truck" />`  
H: None



### Element Attributes

TEXT: Find Your Truck  
TYPE: BUTTON

### Image Annotation

CHOICE: G

### Textual Choices

CHOICE: G

# SeeAct Paradigm

## Action Generation -> Action Grounding

### Action Description

Move the cursor over the "Find Your Truck" button located in the central portion of the webpage, just below the input fields for rental details, and perform a click action.



A: `<text id=0>Moving Trucks & Accessories</text>`  
B: `<div id=1>Where do you want to pick up your truck?</div>`  
C: `<div id=5>When do you want to pick up your truck?</div>`  
... ..  
F: `<input id=18>No value=true</input>`  
G: `<input id=19 button "find your truck" />`  
H: None

### Element Attributes

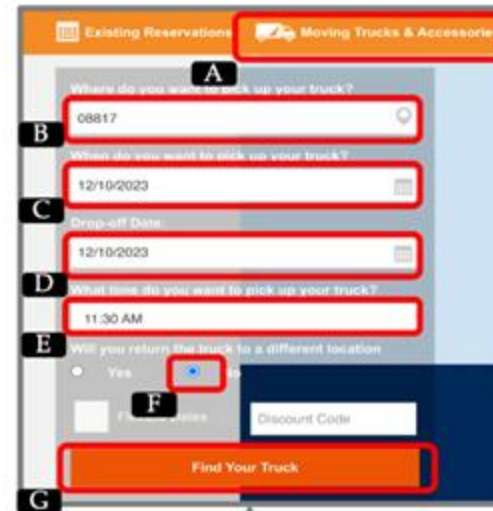
TEXT: Find Your Truck  
TYPE: BUTTON

### Image Annotation

CHOICE: G

### Textual Choices

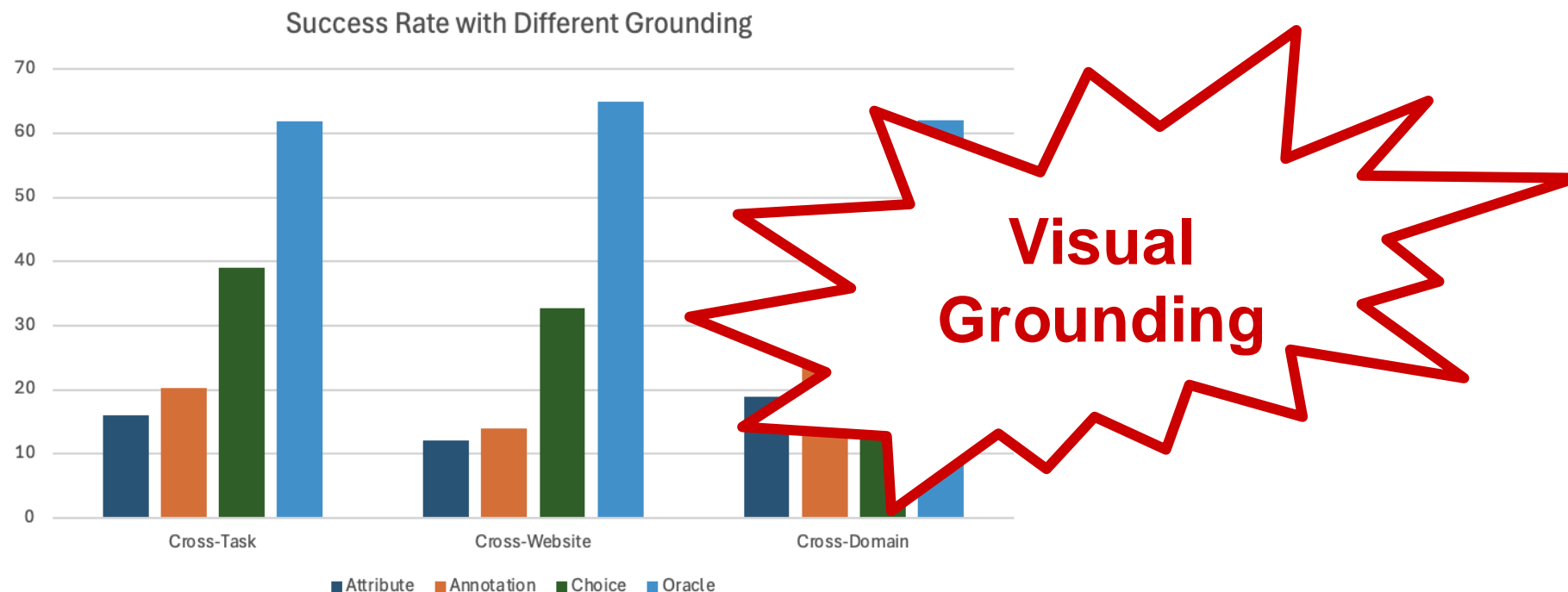
CHOICE: G



# Grounding Strategies

**Oracle:** Ask humans to identify model's intended actions from Action Description

Large Margin between Oracle and SOTA Grounding method so far



Single Step Success Rate on Mind2Web

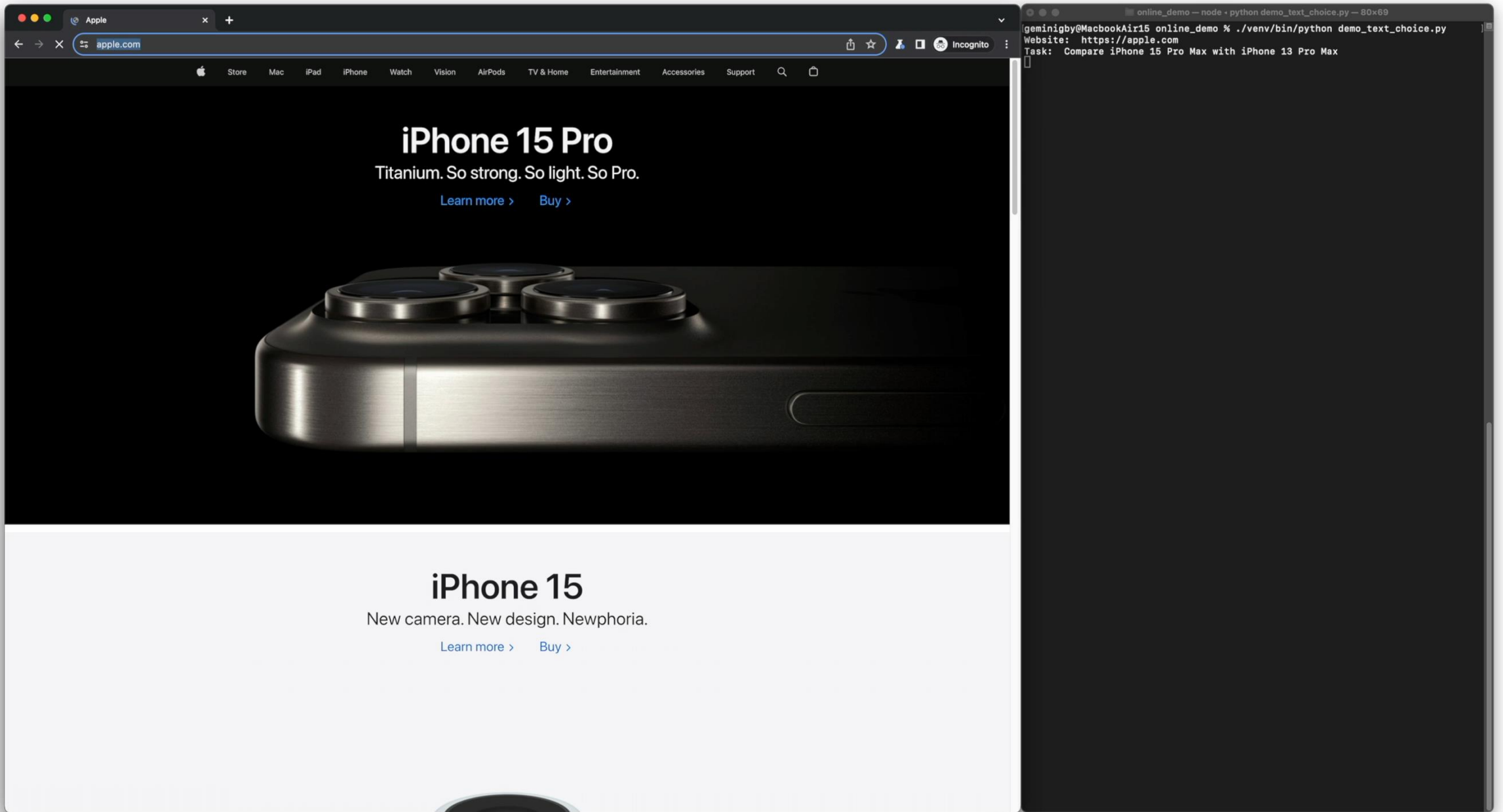
# Human-like Vision-centric Embodiment is the Future for Web/GUI Agents

- Most comprehensive evaluation of GUI Agents to date
- SeeAct-V + UGround outperforms prior SoTA despite its minimalist design



Figure 1: Examples of agent tasks across platforms and performance on **GUI grounding** (♣: ScreenSpot), **offline agent** (♠: Multimodal-Mind2Web, AndroidControl, and OmniAct), and **online agent benchmarks** (♥: Mind2Web-Live and AndroidWorld) when using GPT-4 as the planner.





**Online evaluation tool** that allows SeeAct to interact with live websites



# Online Evaluation

	Offline <sub>0</sub>	Offline <sub>1</sub>	Online
FLAN-T5-XL	4.4	24.4	8.9
GPT-4	1.1	12.2	13.3
SEEACTION <sub>Choice</sub>	3.3	12.2	37.8
SEEACTION <sub>Oracle</sub>	13.3	27.8	51.1

Whole task success rate (%)

# SeeAct Codebase

## An interface between Agent and Website

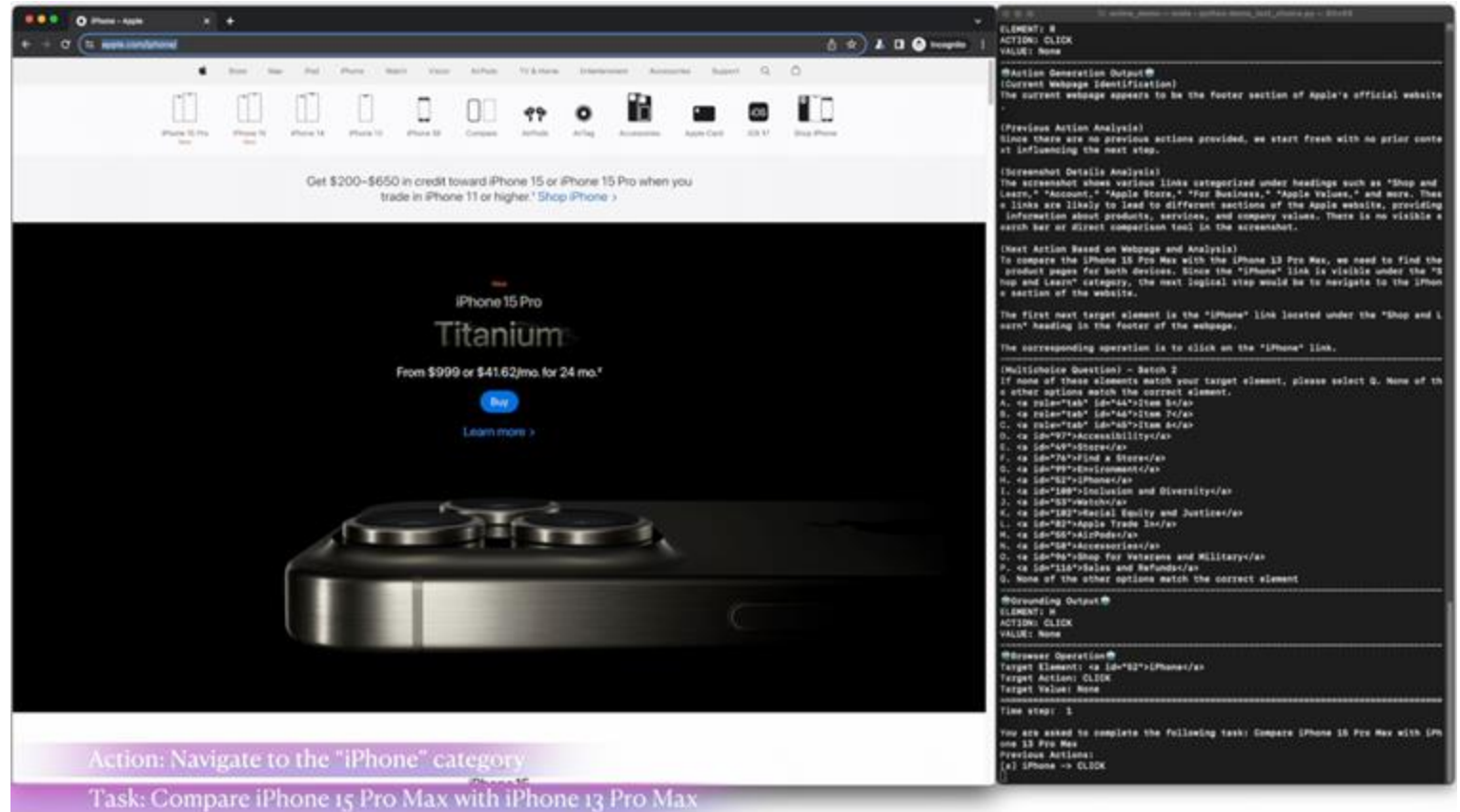
- Perception
- Action Execution

```
import asyncio
import os
from seeact.agent import SeeActAgent

# Setup your API Key here, or pass through environment
os.environ["OPENAI_API_KEY"] = "Your API KEY Here"

async def run_agent():
    agent = SeeActAgent(model="gpt-4-turbo")
    await agent.start()
    while not agent.complete_flag:
        prediction_dict = await agent.predict()
        await agent.execute(prediction_dict)
    await agent.stop()

if __name__ == "__main__":
    asyncio.run(run_agent())
```



SeeAct

654

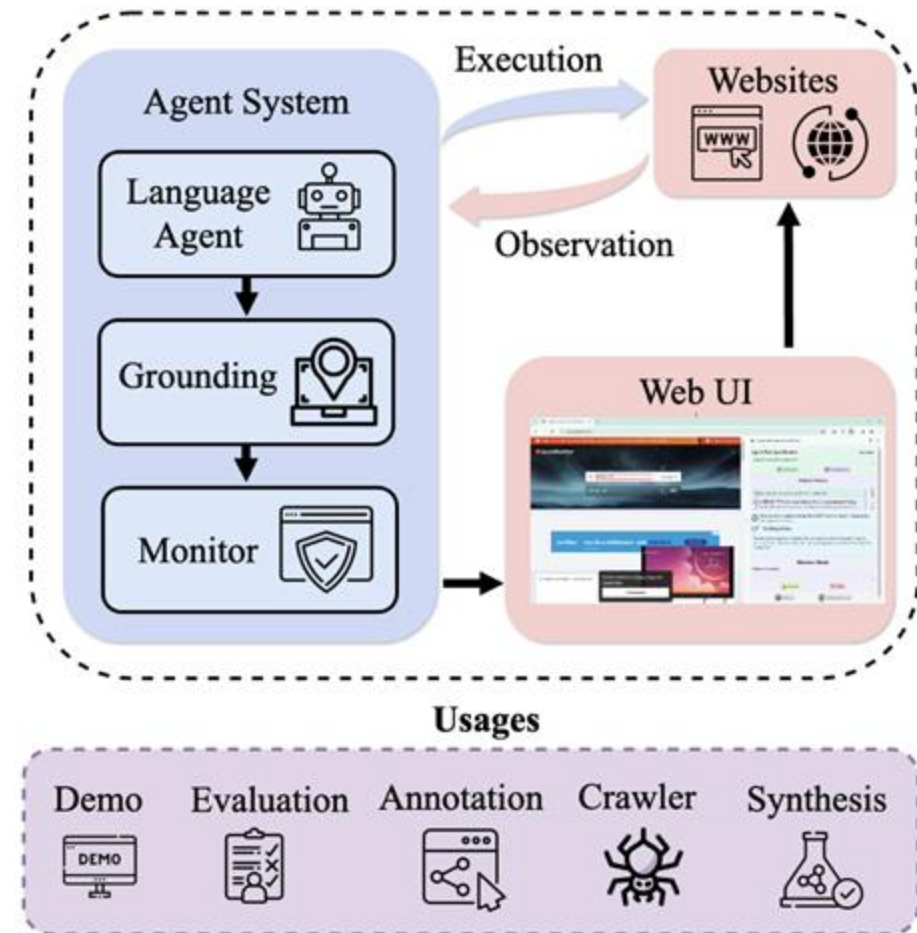
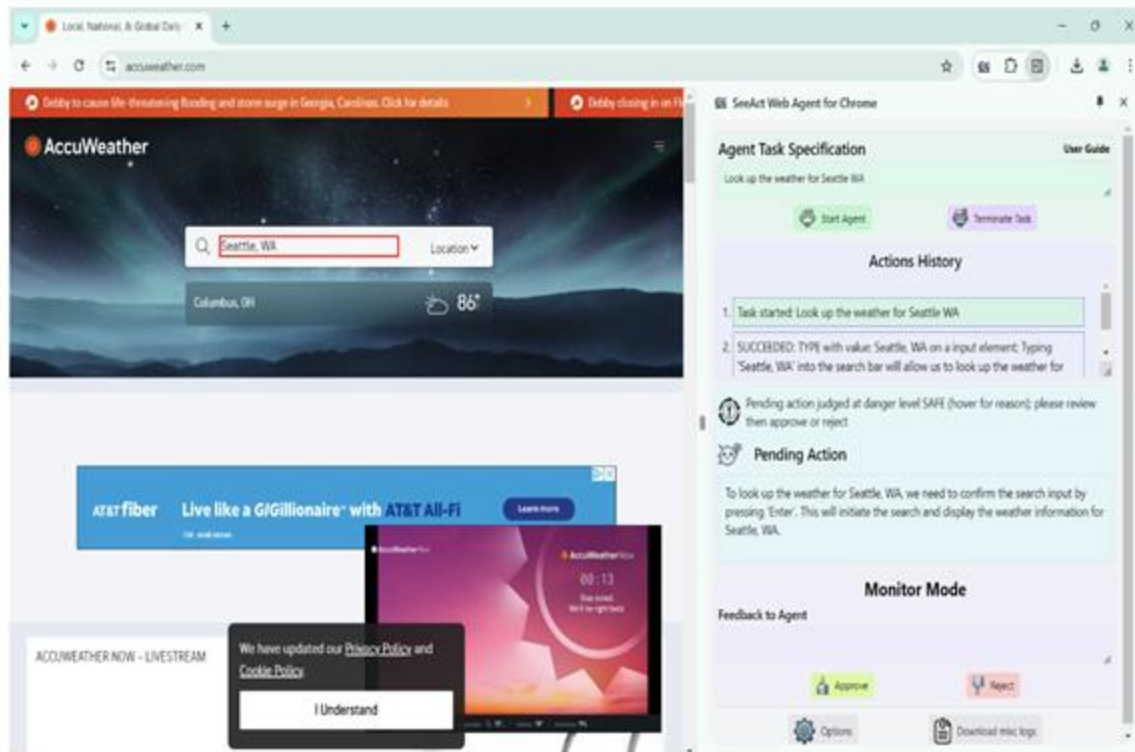


Mind2Web

726

# WebOlympus: An Ecosystem for Web Agent

An ecosystem for web agent

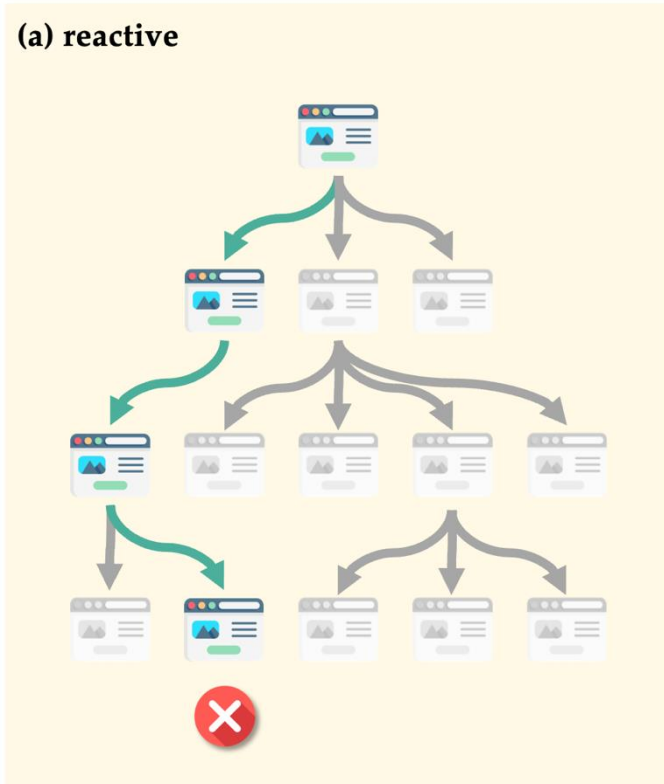


# Outline

- 1 Introduction
- 2 Environment Perception
- 3 Planning**
- 4 WebDreamer: Model-based Planning for Web Agent

# Planning Paradigms for Language Agents

(a) reactive



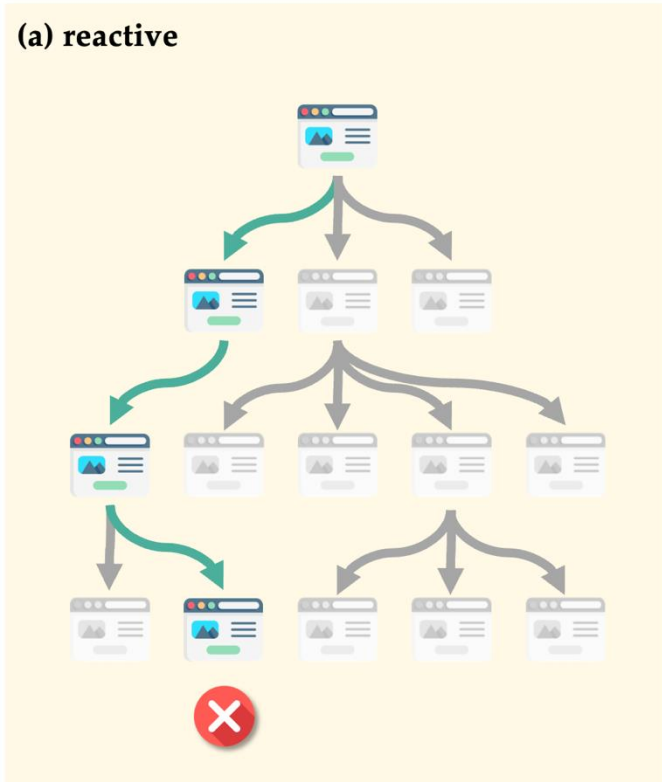
fast, easy to implement



greedy, short-sighted

# Planning Paradigms for Language Agents

(a) reactive

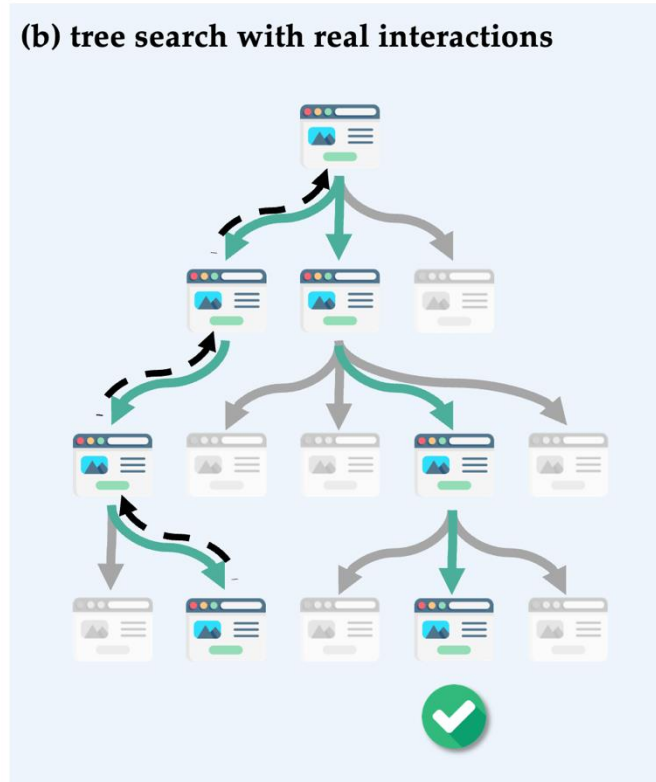


fast, easy to implement



greedy, short-sighted

(b) tree search with real interactions



systematic exploration

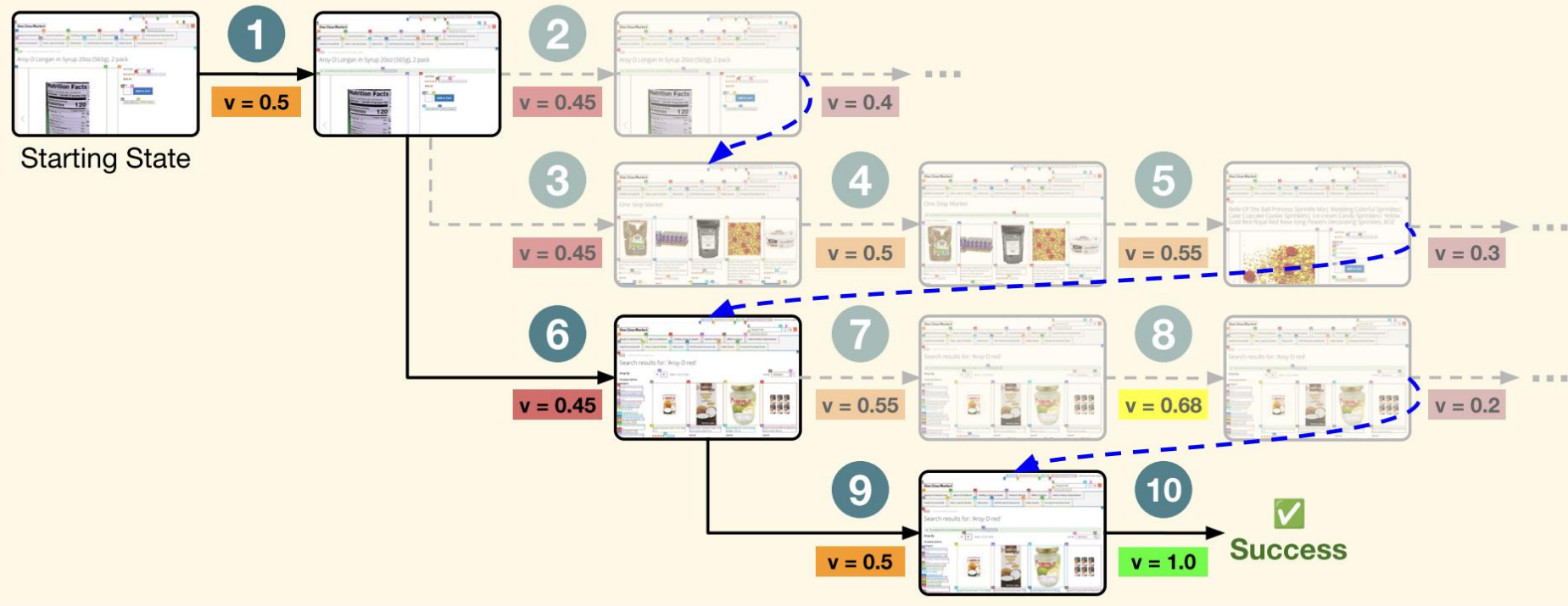


irreversible actions,  
unsafe, slow



# Planning Paradigms for Language Agents

## GPT-4o Agent + Search



- State-Changing Action

# State-Changing Actions

TESLA

Menu

## Demo Drive Model 3

Experience Full Self-Driving (Supervised), Learn About Charging and Get All Your Questions Answered

### Find Location and Time

Zip Code: 95076

Select Location

Gilroy-500 Automall Dr Gilroy CA 500 Automall Dr, CA 95020	<b>SAFE</b> Monterey-Seaside Seaside CA 1901 1901 Del Monte Blvd Seaside, CA 93955, CA 93955	San Jose-Santana Row San Jose CA 333 Santana Row Suite 1015, CA 95128	Su C/ 75 94
---	---	--	----------------------

Date: August 4, 2024 Time: 10:00 AM

### Contact Information

First Name: Steven Last Name: Buckeye

Email Address: steven.zheng010@gmail.com Phone Number: US +1 (443) 469-6056

☒ Learn about Energy Products

**MEDIUM**

[Schedule Demo Drive](#)

AVIS

AVIS PREFERRED WELCOME, BOYUAN MENU

## ADDRESS & PHONE NUMBER

Save up to 2 addresses and 2 phone numbers.

Primary Address

Country ⓘ  
USA

Address Line 1  
281 W Lane Ave, Columbus

Address Line 2 (optional)

Zipcode: 43210 City: COLUMBUS State/Province: OH

**HIGH**

[CANCEL](#) [SAVE](#)

**LOW**

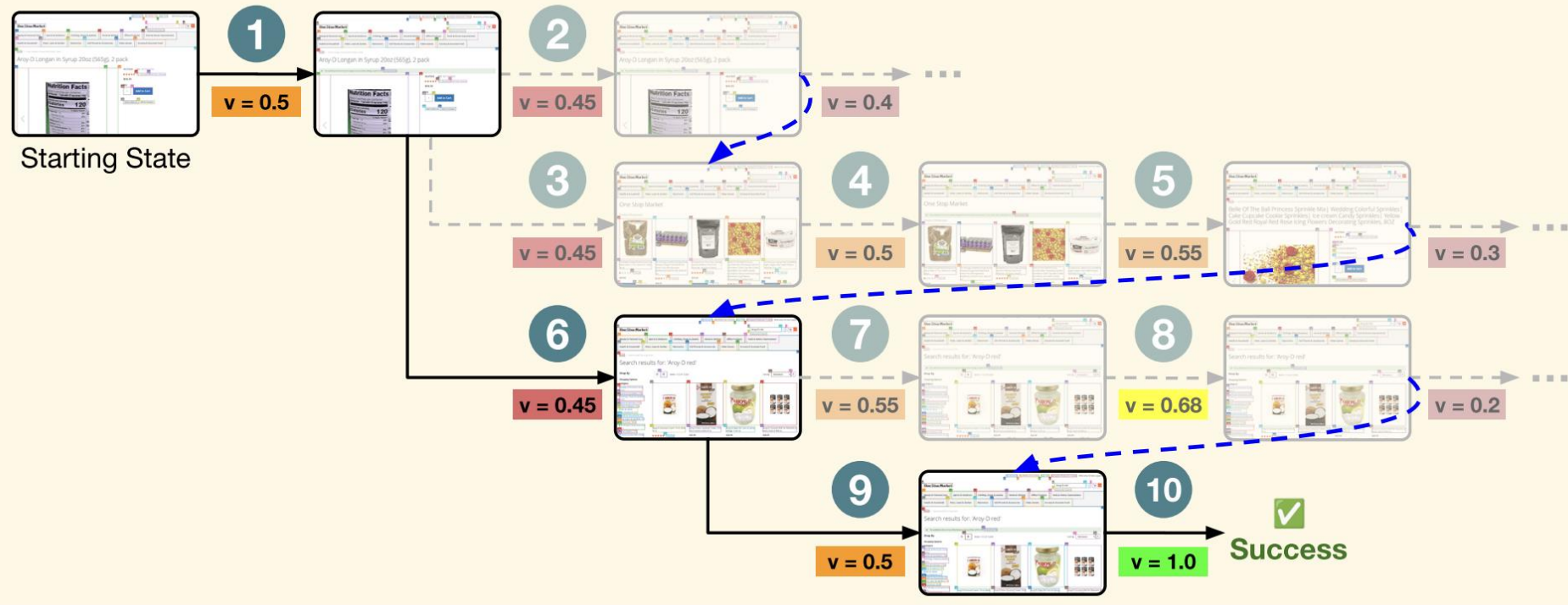
**AVIS** We and our partners use cookies and other technologies to collect your information and interactions so that we can improve your experience (see our [Privacy Policy](#) and [Terms of Use](#)).

[Your Privacy Choices](#)

[Accept All Cookies](#)

# Planning Paradigms for Language Agents

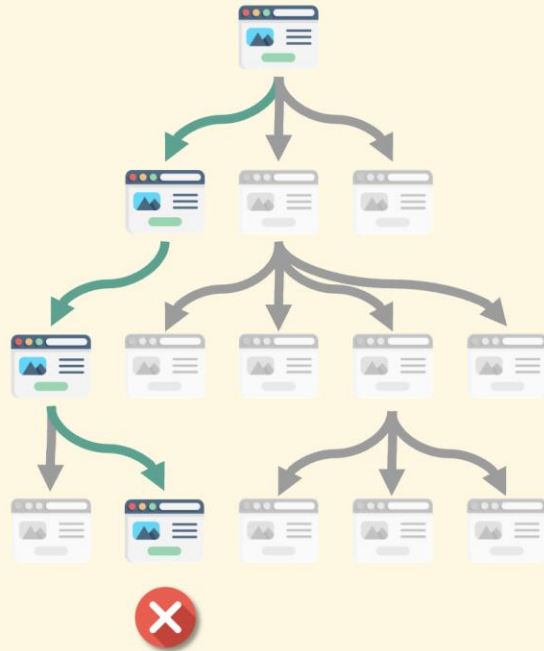
## GPT-4o Agent + Search



- State-Changing Action
- Tracing Back?
- Limited Efficiency

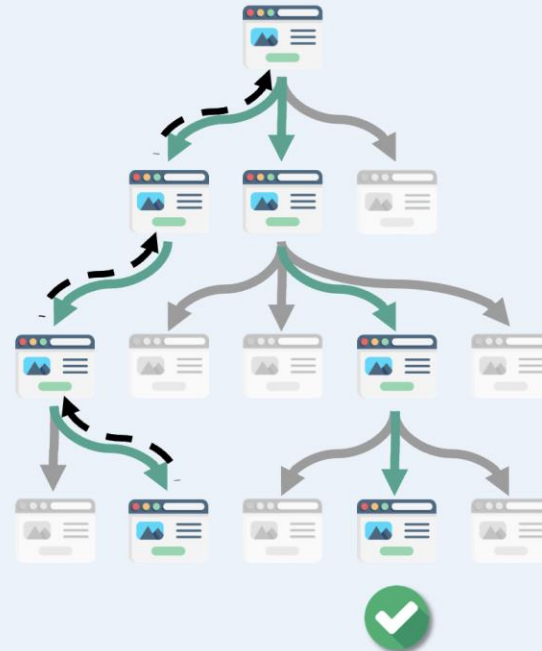
# Planning Paradigms for Language Agents

(a) reactive



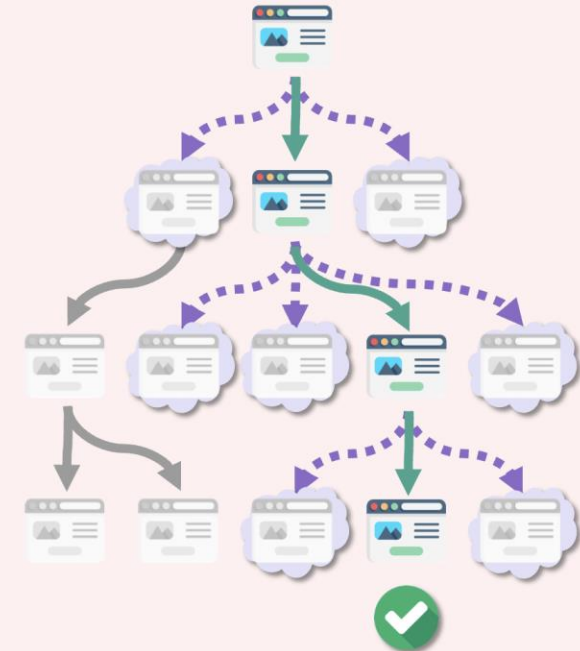
- 😊 fast, easy to implement
- 😐 greedy, short-sighted

(b) tree search with real interactions



- 😊 systematic exploration
- 😐 irreversible actions, unsafe, slow

(c) model-based planning



- 😊 faster, safer, systematic exploration
- 😐 how to get a world model

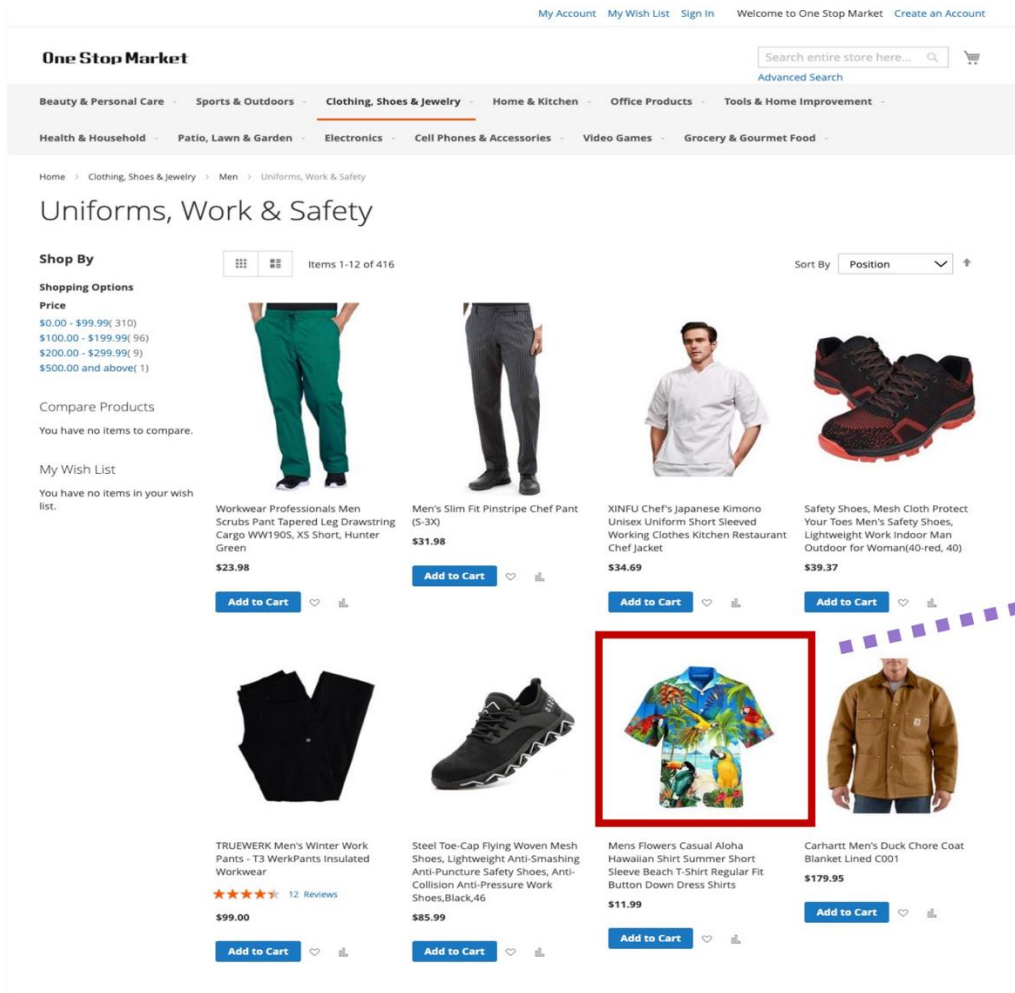
# World Model

A model capable of predicting environment transition:

$$\hat{T}: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$$

If I do this ( $a_t$ ) on ( $s_t$ ), what would happen next ( $s_{t+1}$ )?

# LLM can predict state transitions(reasonably good)

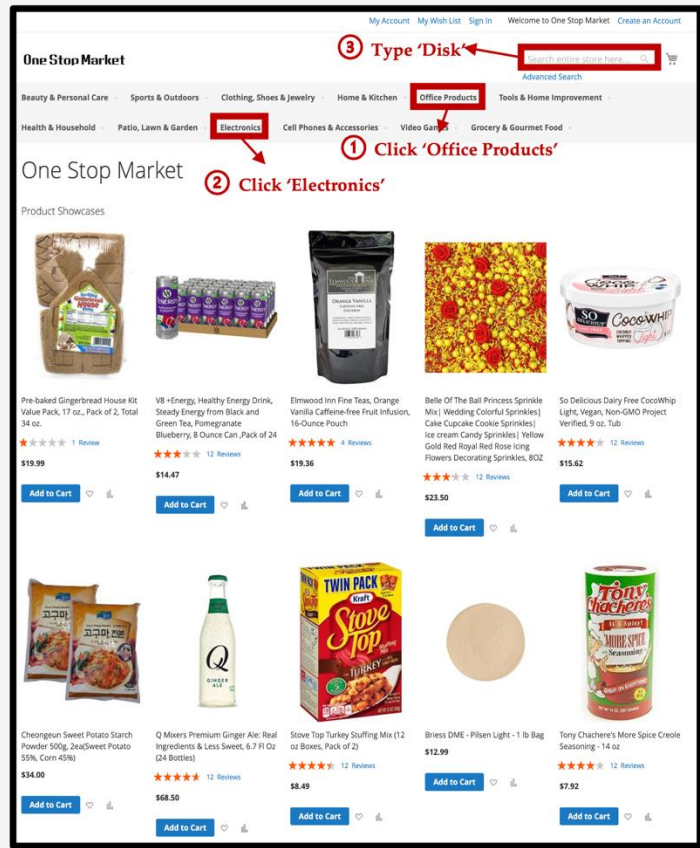


The page will navigate to a detailed product page for the "Mens Flowers Casual Aloha Hawaiian Shirt Summer Short Sleeve Beach T-Shirt Regular Fit Button Down Dress Shirts." This new page will likely contain additional information about the product including more detailed specifications, customer reviews, larger images, sizing options, and possibly a larger "Add to Cart" button. Other elements from the current category view like the grid of products will be replaced with the detailed view of this specific product.



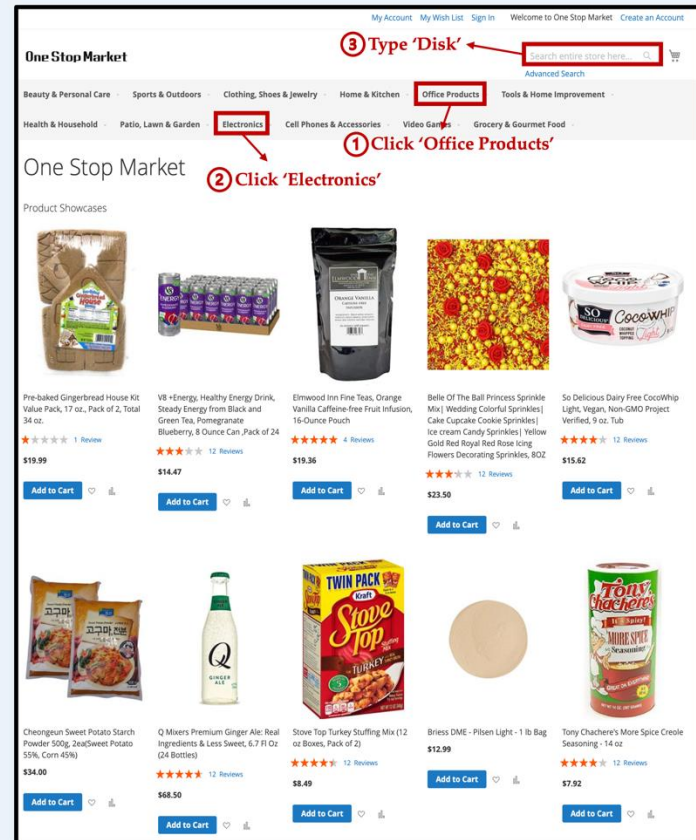
# WebDreamer: model-based planning for web agents

Please navigate to the 'Data Storage' category and purchase the least expensive disk with 512GB of storage.

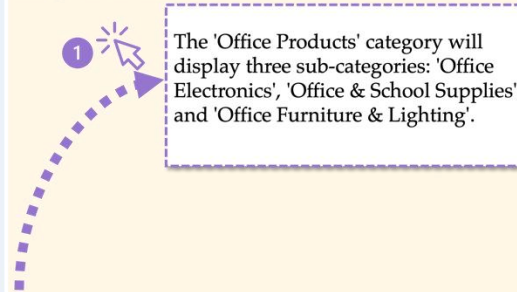


# WebDreamer: model-based planning for web agents

Please navigate to the 'Data Storage' category and purchase the least expensive disk with 512GB of storage.

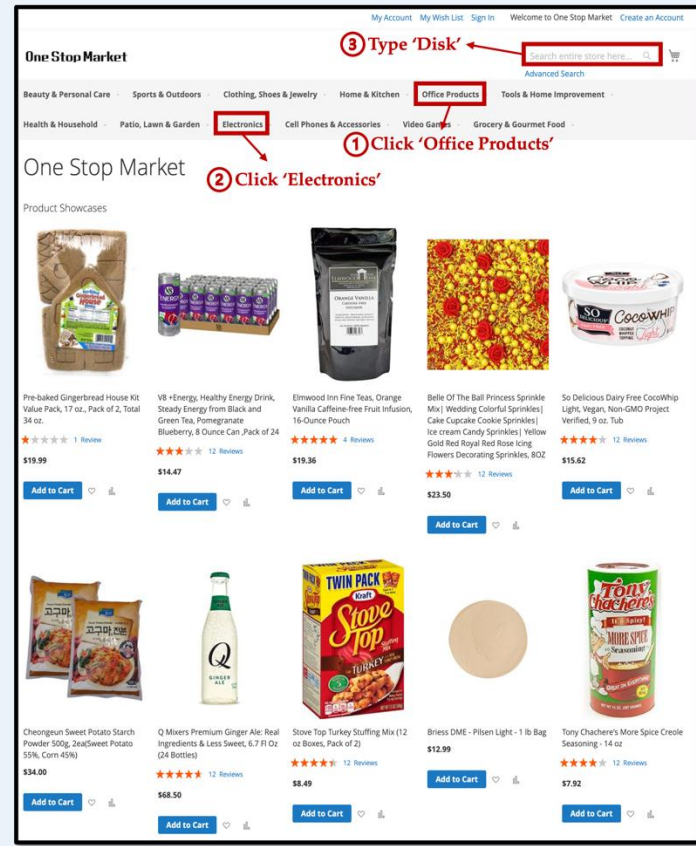


## Stage I: Simulation

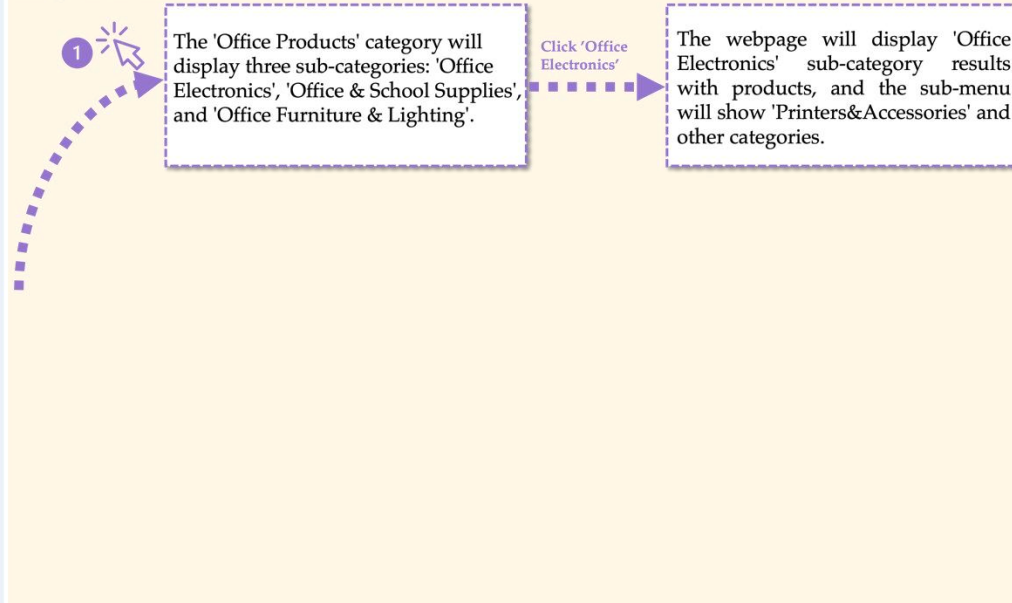


# WebDreamer: model-based planning for web agents

Please navigate to the 'Data Storage' category and purchase the least expensive disk with 512GB of storage.

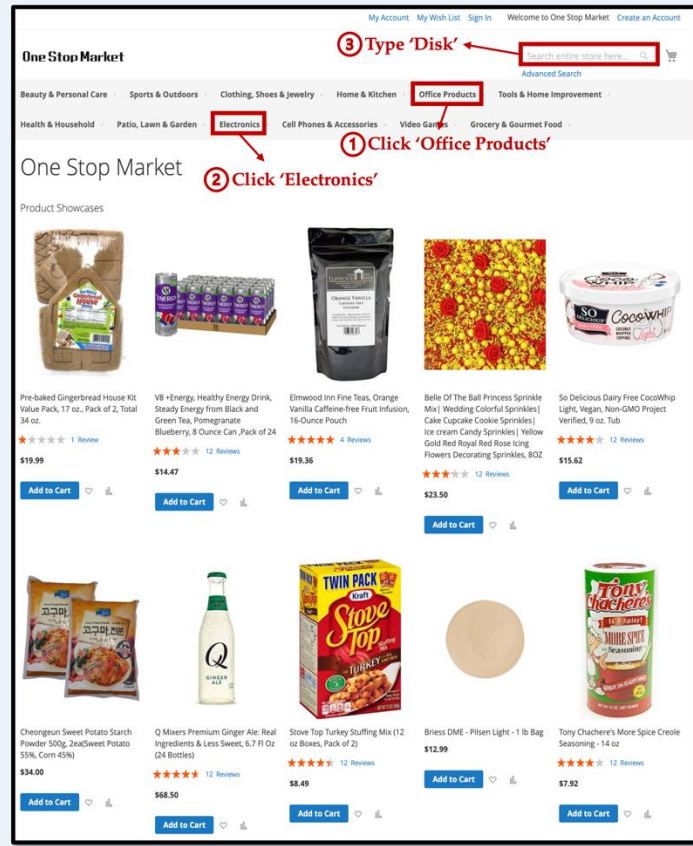


## Stage I: Simulation

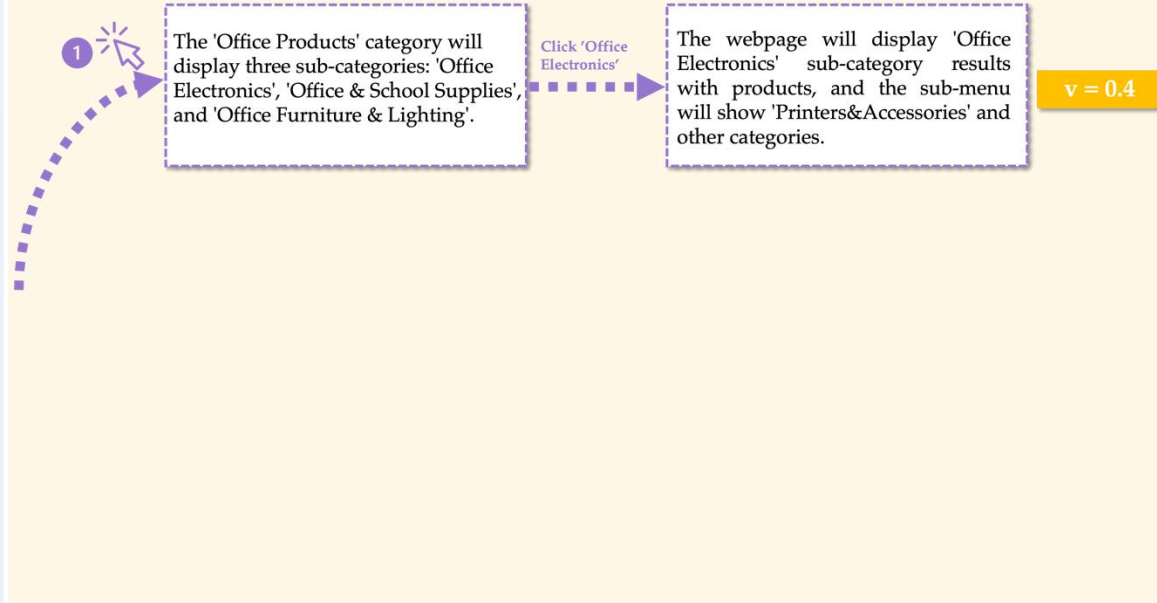


# WebDreamer: model-based planning for web agents

Please navigate to the 'Data Storage' category and purchase the least expensive disk with 512GB of storage.



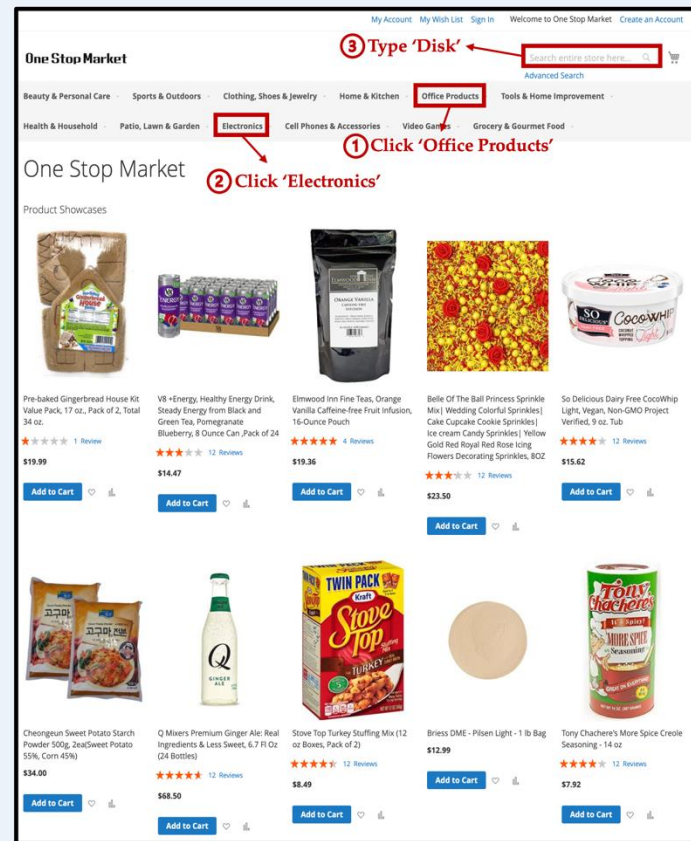
## Stage I: Simulation



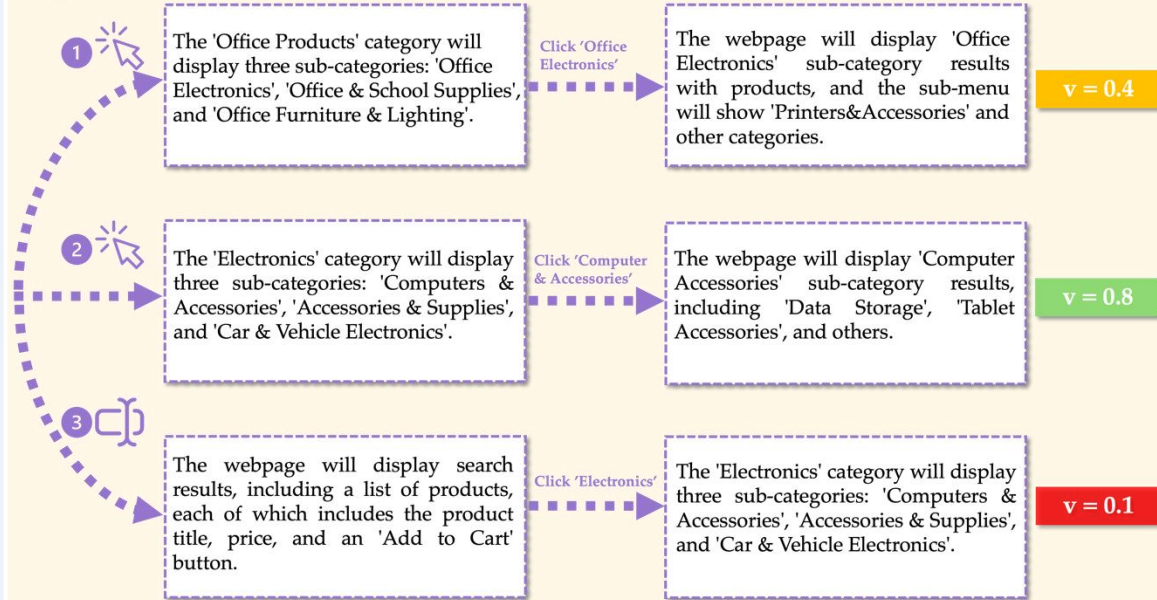


# WebDreamer: model-based planning for web agents

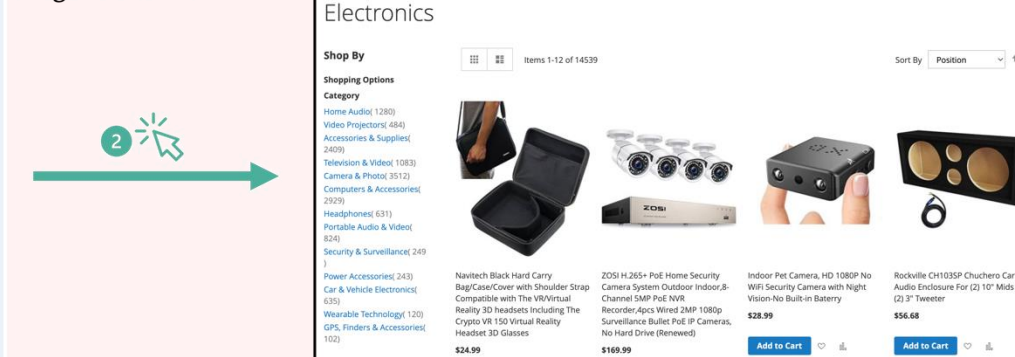
Please navigate to the 'Data Storage' category and purchase the least expensive disk with 512GB of storage.



## Stage I: Simulation



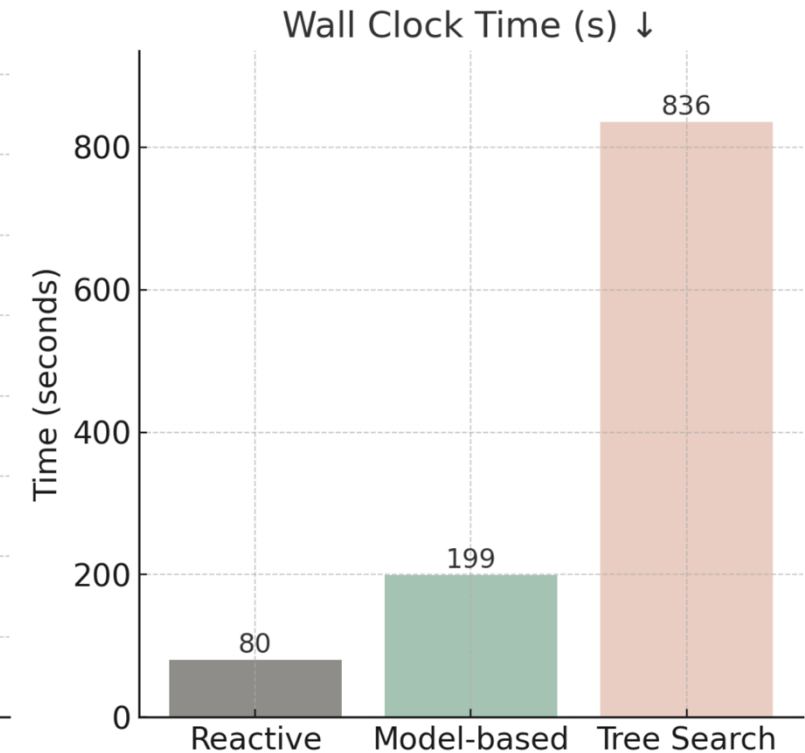
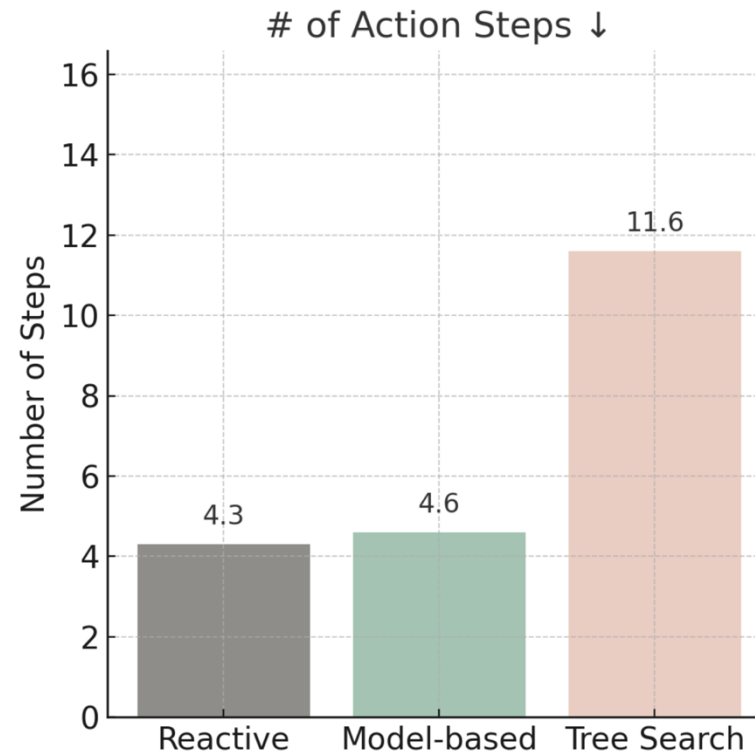
## Stage II: Execution





# Results on VisualWebArena

Model-based planning is more accurate than reactive planning and more efficient than tree search



# Outline

- 1 Introduction
- 2 Environment Perception
- 3 Planning
- 4 Self-Improvement**

What's next step?

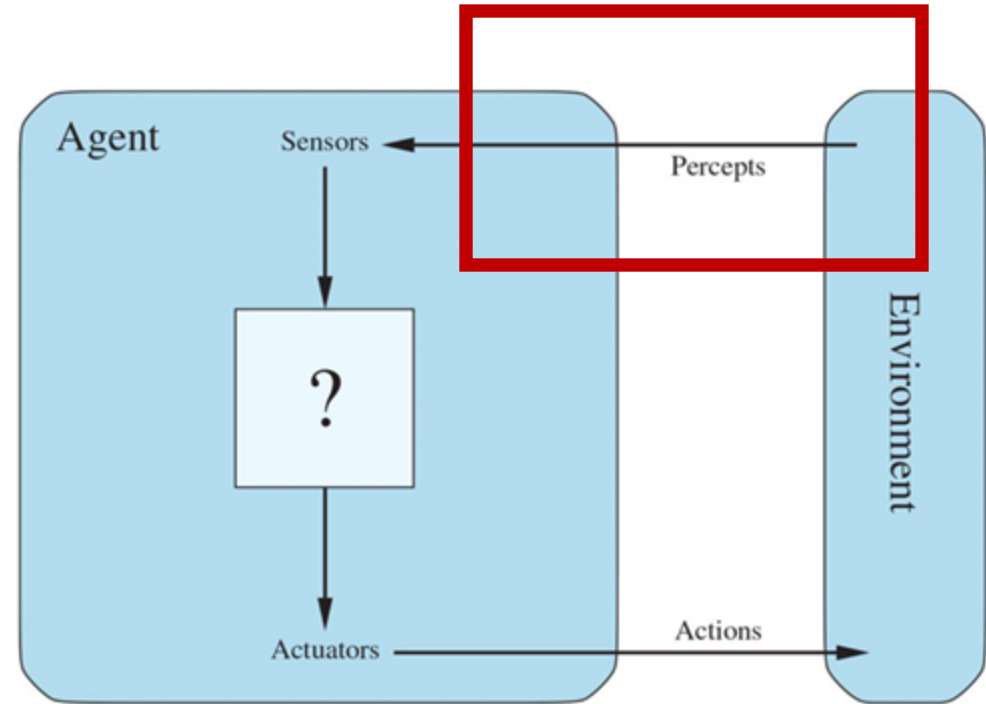
Where have we arrived at?

# Perception

Perception:

HTML/DOM: MindAct

Screenshot: SeeAct, WebGUM, CogAgent

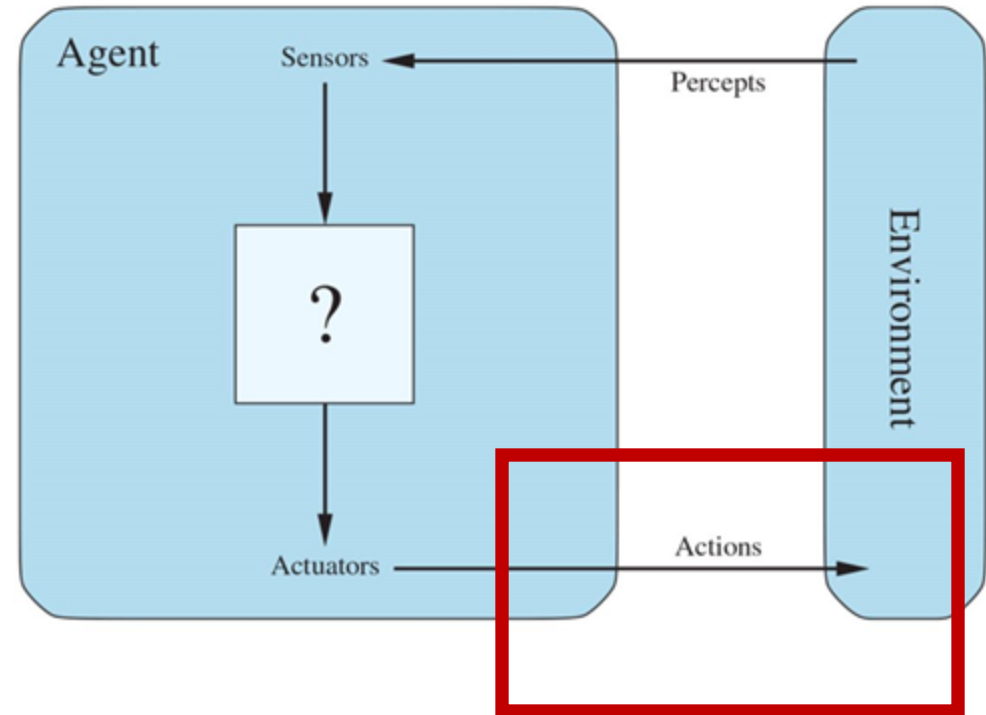
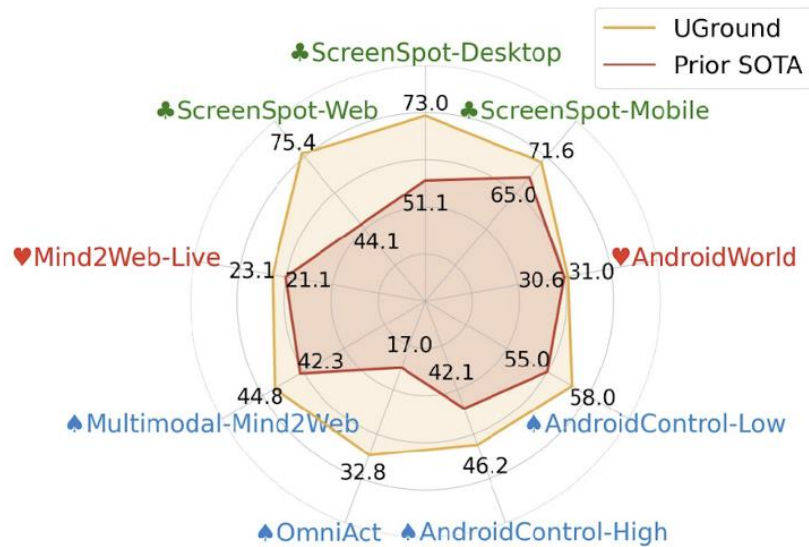


# Action

Action:

Weak Grounding: SOM, SeeAct

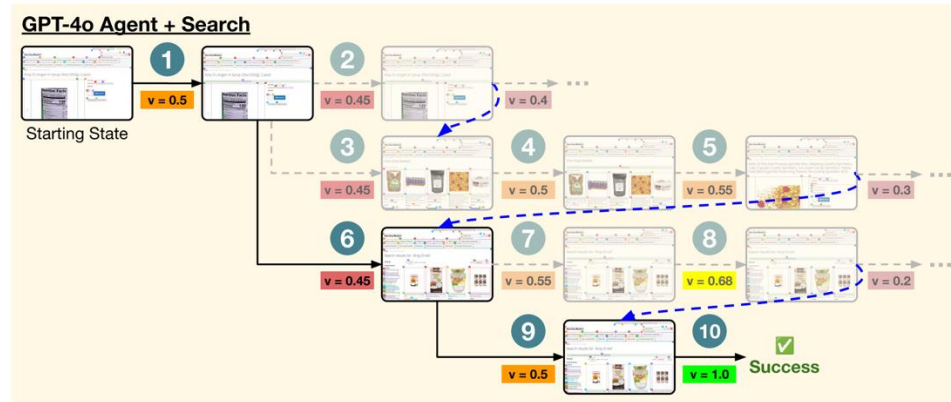
Pixel-Level: UGround, SeeClick, etc.



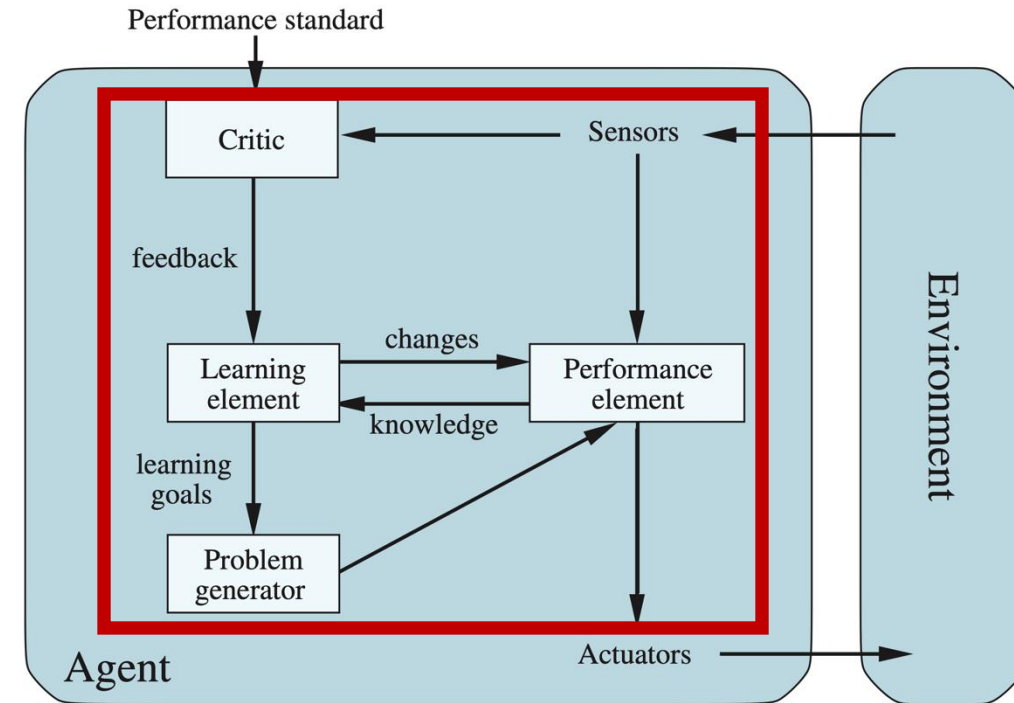
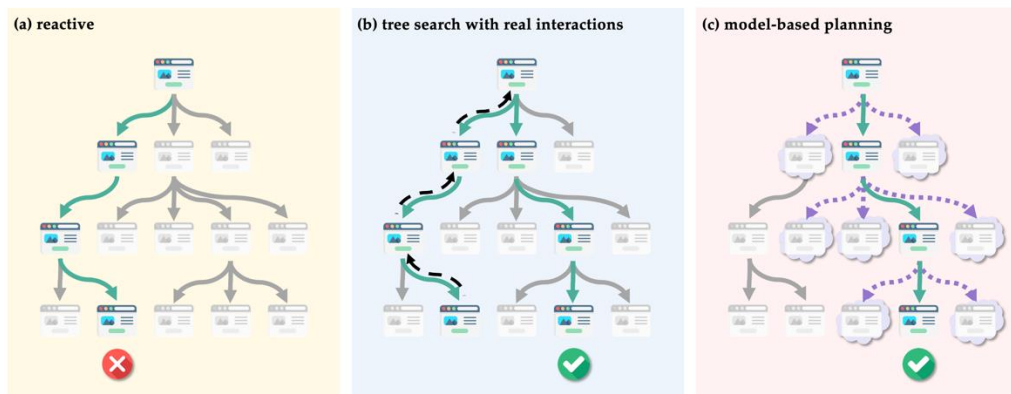


# Planning

## Search over the environment: Search-Agent



## Planning with World Model: WebDreameer



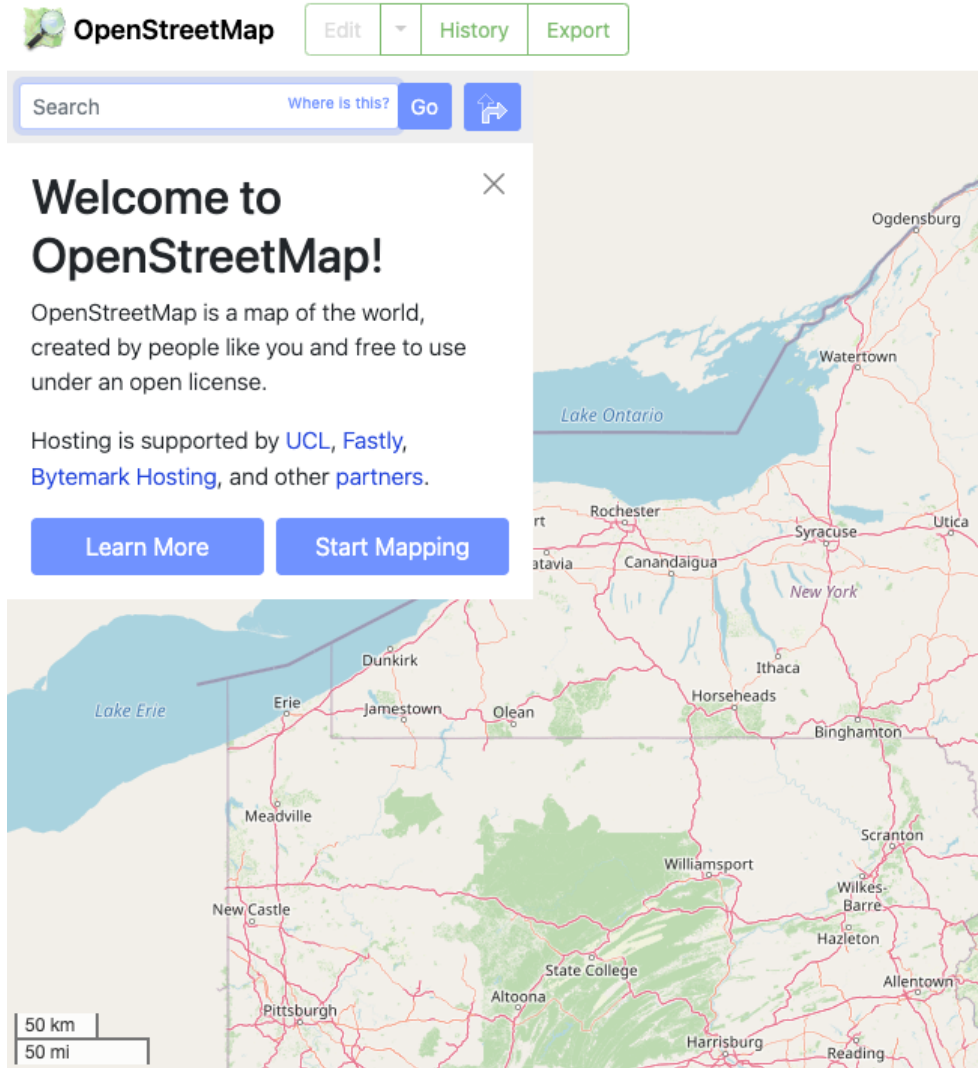
# What's next step?

Agent has reached a decent performance of 30~50%.

How reach closer to 100% to be actually useful and robust?

What current agents are capable of?

# Task Proposal



## 1. **\*\*Get Directions Between Two Points:\*\***

- Usefulness: 5 (Finding directions is a frequent task for users of a map service.)
- Steps: Click on 'Find directions between two points' link, enter starting point, enter destination, click 'Go'. Total: 4 actions.
- Total Rating: 9

## 2. **\*\*Export Map Data:\*\***

- Usefulness: 4 (Exporting map data is useful for offline usage or data analysis.)
- Steps: Click on 'Export' link, select export options, click 'Export'. Total: 3 actions.
- Total Rating: 7

## 3. **\*\*Search for Location:\*\***

- Usefulness: 5 (Common and frequent usage of map services.)
- Steps: Enter text into 'Search' textbox, click 'Go'. Total: 2 actions.
- Total Rating: 7

## 4. **\*\*Show My Location:\*\***

- Usefulness: 4 (Useful for quickly finding your current location on the map.)
- Steps: Click 'Show My Location' button. Total: 1 action.
- Total Rating: 5

The skill 'Get Directions Between Two Points' has the highest total rating and usefulness, making it the most beneficial skill to propose for this task.

# Execution

**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>`

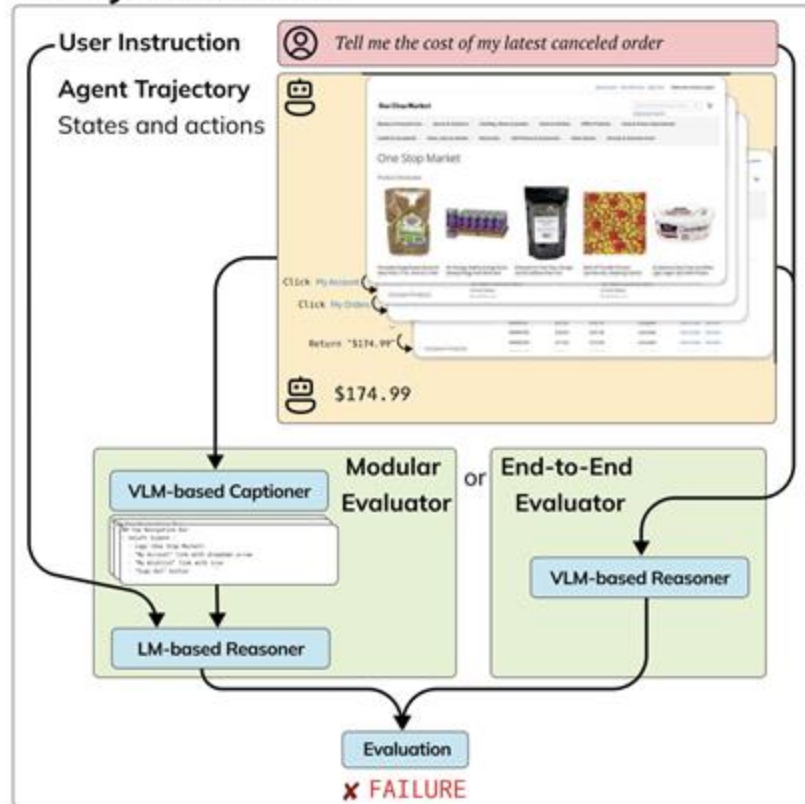


`<a href="link:XXX">Read  
Reviews</a>`

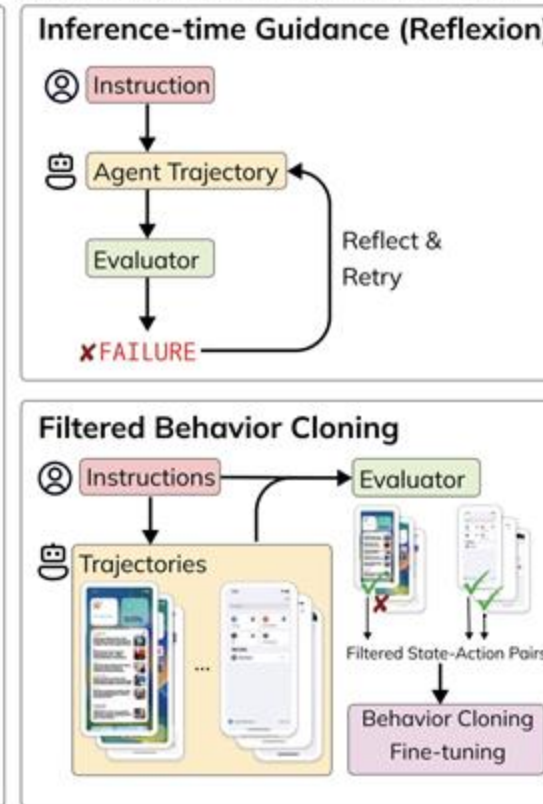
# Critic

Given action sequence and instruction, label if the task is successfully completed

## Policy Evaluation



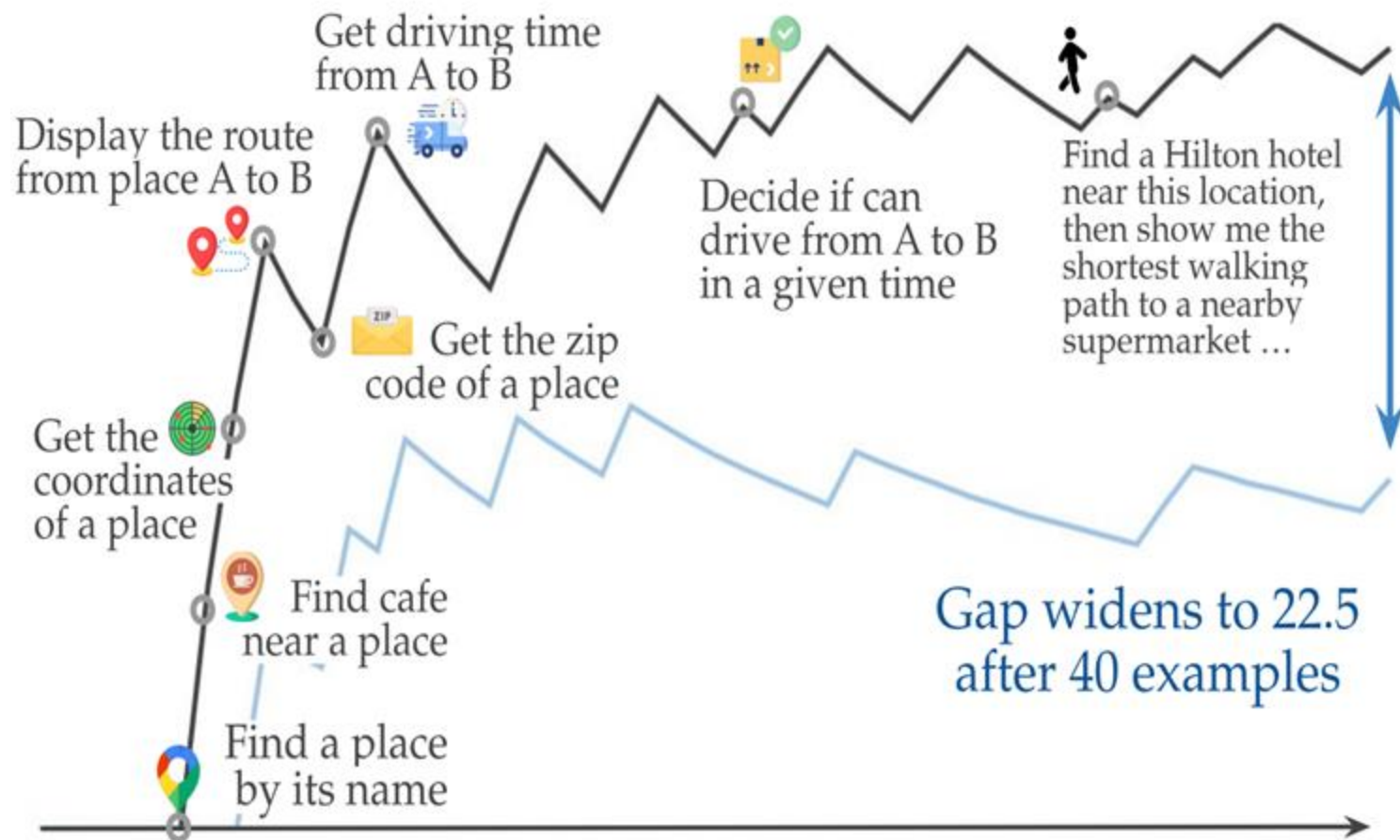
## Autonomous Refinement





# Self-Improvement through exploration

1. Propose tasks
2. Explore websites
3. Accumulate skills
4. Leverage knowledge



# What's the Abstraction Level?

Trajectories: Accumulate knowledge in an inexplicit way

- Heavy: Training Policy
- Update & Adapt to new websites? Catastrophic forgetting
- Website changes?

Workflow: Textual Description of Procedure

- Lengthy
- How to update workflow?
- How to leverage multiple attempts into a workflow?

# What's the Abstraction Level?

## APIs:

- Light-Weighted
- Easy to Debug
- Easy for trail-and-error

```
async def identify_pill(page, imprint, color=None, shape=None):
    """
    Automates the process of identifying a pill using the Pill Identifier
    feature on Drugs.com.

    Parameters:
    - page: The Playwright page object.
    - imprint: The imprint on the pill to be identified.
    - color: (Optional) The color of the pill.
    - shape: (Optional) The shape of the pill.

    This function navigates to the Pill Identifier page, agrees to the terms,
    inputs the pill's characteristics,
    and submits the information for identification.

    Usage Log:
    - Successfully navigated to the Pill Identifier page and submitted pill
    information for identification.
    - Inputted imprint '93 5510', color 'White', and shape 'Oval' and
    successfully submitted for identification.
    - Encountered issues with strict mode violations when attempting to click
    the 'Search' button due to multiple matches.
    - Updated to use a more specific selector for the 'Search' button to avoid
    strict mode violations.
    """
    import re

    await page.goto("https://www.drugs.com/pill_identification.html")
    await page.get_by_role("link", name="Agree and Continue").click()
    await page.get_by_role("text", name="Pill Imprint").fill(imprint)
    if color:
        await page.get_by_role("group", name="Color and shape (optional)").
            get_by_role("combobox", name="Color
            (optional)").select_option(color)
    if shape:
        await page.get_by_role("group", name="Color and shape (optional)").
            get_by_role("combobox", name="Shape
            (optional)").select_option(shape)

    search_button =
        page.locator("button.ddc-btn.ddc-btn-block[data-submit-loading]")
    await search_button.click()
```

# What if?

We have an algorithm:

Explore environment and accumulate skill&knowledge into APIs

A plug-and-play module for any agent capable of function calling

# Overview

## Skill Library



check\_drug\_interaction(drug\_name)



subscribe\_to\_newsletters(email)



search\_FDA\_alert()



check\_side\_effects(drug\_name)



identify\_pill(imprint, color)

## Skill Synthesis

Practice Skill



Reward Model



API Synthesis

```
async def identify_pill(page, imprint, color):
    # Automates the process of identifying a pill on Drugs.com.
    await page.goto("/pill_identification.html")
    await page.get_by_role("link", name="Agree and Continue").click()
    await page.get_by_role("textbox", name="Pill Imprint").fill(imprint)
    search_button = page.get_by_role("form", name="Pill Identifier").get_by_role("button", name="Search")
    + if color: await page.get_by_role("group", name="Color and shape (optional)").get_by_role(
    + "combobox", name="Color (optional)").select_option(color)
    await search_button.click()
```

## Verification

Test Cases



Execution



### Error:

Warning: Unused parameter 'color'  
-> identify\_pill(page, imprint, color)  
The parameter 'color' is defined but never used in the function body

Error: Search results validation failed.  
API returned pills matching imprint '5510' but the wrong color (expected: yellow).

### Diagnosis:

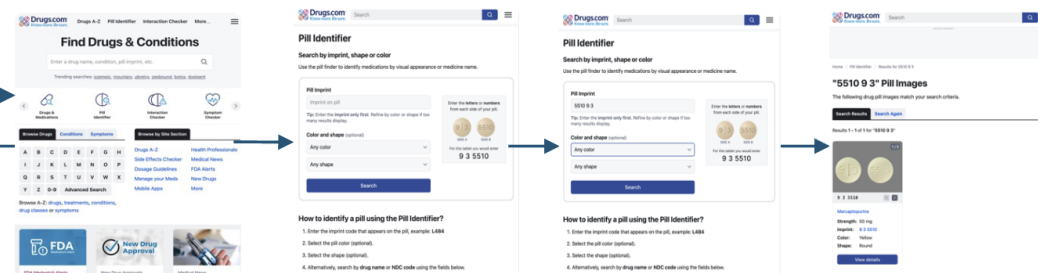
Added color selection using the dropdown menu  
Made color filtering optional

Uses 'if color': to make color filtering optional  
Properly implemented color parameter in the search function, locating the "Color and shape" group element

## Skill Proposal

	time	useful	
Identify Pill Using Pill Identifier	3	5	✓
Check Side Effects for a Drug	3	4	✗
Search a Condition's Treatment	3	4	✗
Browse Drugs by Letter	2	4	✗

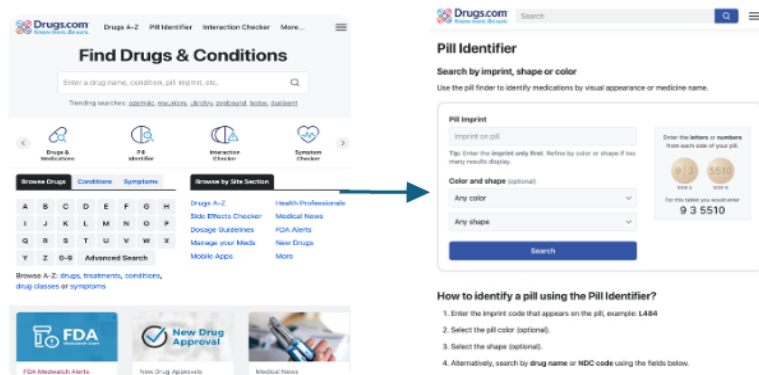
## Environment










# Skill Proposal

## Environment



## Skill Library

-  `check_drug_interaction(drug_name)`
-  `subscribe_to_newsletters(email)`
-  `search_FDA_alert()`
-  `check_side_effects(drug_name)`
-  `identify_pill(imprint, color)`

## Skill Proposal

	time	useful	
Identify Pill Using Pill Identifier	3	5	✓
Check Side Effects for a Drug	3	4	✗
Search a Condition's Treatment	3	4	✗
Browse Drugs by Letter	2	4	✗

# Skill Synthesis

## Skill Library



check\_drug\_interaction(drug\_name)



subscribe\_to\_newsletters(email)



search\_FDA\_alert()



check\_side\_effects(drug\_name)



identify\_pill(imprint, color)

## Skill Synthesis

**Practice Skill**



**Reward Model**



**API Synthesis**

```
async def identify_pill(page, imprint, color):  
    # Automates the process of identifying a pill on Drugs.com.  
    await page.goto("/pill_identification.html")  
    await page.get_by_role("link", name="Agree and Continue").click()  
    await page.get_by_role("textbox", name="Pill Imprint").fill(imprint)  
    search_button = page.get_by_role("form", name="Pill Identifier").get_by_role("button", name="Search")  
  
    await search_button.click()
```

# Execution

**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>`

# Skill Synthesis

## Skill Library



check\_drug\_interaction(drug\_name)



subscribe\_to\_newsletters(email)



search\_FDA\_alert()



check\_side\_effects(drug\_name)



identify\_pill(imprint, color)

## Skill Synthesis

Practice Skill



Reward Model



API Synthesis

```
async def identify_pill(page, imprint, color):  
    # Automates the process of identifying a pill on Drugs.com.  
    await page.goto("/pill_identification.html")  
    await page.get_by_role("link", name="Agree and Continue").click()  
    await page.get_by_role("textbox", name="Pill Imprint").fill(imprint)  
    search_button = page.get_by_role("form", name="Pill Identifier").get_by_role("button", name="Search")  
  
    await search_button.click()
```

# Skill Synthesis

## Skill Library



check\_drug\_interaction(drug\_name)



subscribe\_to\_newsletters(email)



search\_FDA\_alert()



check\_side\_effects(drug\_name)



identify\_pill(imprint, color)

## Skill Synthesis

Practice Skill



Reward Model



API Synthesis

```
async def identify_pill(page, imprint, color):  
    # Automates the process of identifying a pill on Drugs.com.  
    await page.goto("/pill_identification.html")  
    await page.get_by_role("link", name="Agree and Continue").click()  
    await page.get_by_role("textbox", name="Pill Imprint").fill(imprint)  
    search_button = page.get_by_role("form", name="Pill Identifier").get_by_role("button", name="Search")  
  
    await search_button.click()
```

# Verification

```
async def identify_pill(page, imprint, color):  
    # Automates the process of identifying a pill on Drugs.com.  
    await page.goto("/pill_identification.html")  
    await page.get_by_role("link", name="Agree and Continue").click()  
    await page.get_by_role("textbox", name="Pill Imprint").fill(imprint)  
    search_button = page.get_by_role("form", name="Pill Identifier").get_by_role("button", name="Search")  
  
    await search_button.click()
```



## Pill Identifier

### Search by imprint, shape or color

Use the pill finder to identify medications by visual appearance or medicine name.

#### Pill Imprint

Tip: Enter the imprint only first. Refine by color or shape if too many results display.

#### Color and shape (optional)

Enter the letters or numbers from each side of your pill.



For this tablet you would enter

9 3 5510

### How to identify a pill using the Pill Identifier?

1. Enter the imprint code that appears on the pill, example: **L484**
2. Select the pill color (optional).
3. Select the shape (optional).
4. Alternatively, search by **drug name** or **NDC code** using the fields below.



# Verification

```
async def identify_pill(page, imprint, color):  
    # Automates the process of identifying a pill on Drugs.com.  
    await page.goto("/pill_identification.html")  
    await page.get_by_role("link", name="Agree and Continue").click()  
    await page.get_by_role("textbox", name="Pill Imprint").fill(imprint)  
    search_button = page.get_by_role("form", name="Pill Identifier").get_by_role("button", name="Search")  
  
    await search_button.click()
```

Pill Imprint

Tip: Enter the imprint only first. Refine by color or shape if too many results display.

Color and shape (optional)

Any color

Any shape

Search

Enter the letters or numbers from each side of your pill.

9 | 3

5510

SIDE A

SIDE B

For this tablet you would enter

9 3 5510

## Verification

Test Cases



Execution



### Error:

Warning: Unused parameter 'color'

-> identify\_pill(page, imprint, color)

The parameter 'color' is defined but never used in the function body

Error: Search results validation failed.

API returned pills matching imprint '5510' but the wrong color (expected: yellow).

### Diagnosis:

Added color selection using the dropdown menu

Made color filtering optional

Uses 'if color': to make color filtering optional  
Properly implemented color parameter in the search function, locating the "Color and shape" group element

# Verification

```
async def identify_pill(page, imprint, color):  
    # Automates the process of identifying a pill on Drugs.com.  
    await page.goto("/pill_identification.html")  
    await page.get_by_role("link", name="Agree and Continue").click()  
    await page.get_by_role("textbox", name="Pill Imprint").fill(imprint)  
    search_button = page.get_by_role("form", name="Pill Identifier").get_by_role("button", name="Search")  
    + if color: await page.get_by_role("group", name="Color and shape (optional)").get_by_role(  
    + "combobox", name="Color (optional)").select_option(color)  
    await search_button.click()
```

Pill Imprint

Tip: Enter the imprint only first. Refine by color or shape if too many results display.

Color and shape (optional)

Any color

Any shape

Search

Enter the letters or numbers from each side of your pill.

9 | 3

5510

SIDE A

SIDE B

For this tablet you would enter

9 3 5510

## Verification

### Test Cases

### Execution



#### Error:

Warning: Unused parameter 'color'

-> identify\_pill(page, imprint, color)

The parameter 'color' is defined but never used in the function body

Error: Search results validation failed.

API returned pills matching imprint '5510' but the wrong color (expected: yellow).

#### Diagnosis:

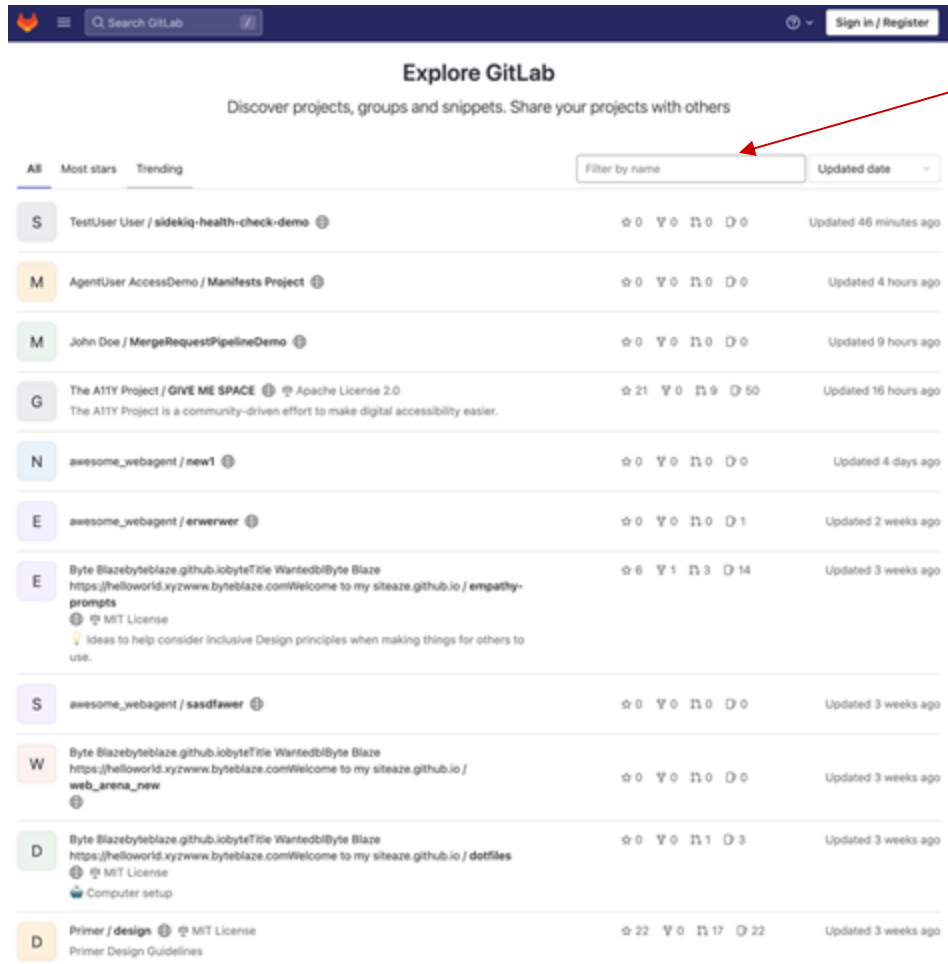
Added color selection using the dropdown menu

Made color filtering optional

Uses 'if color': to make color filtering optional  
Properly implemented color parameter in the search function, locating the "Color and shape" group element

# More Examples of Synthesized APIs

## APIs in GitLab



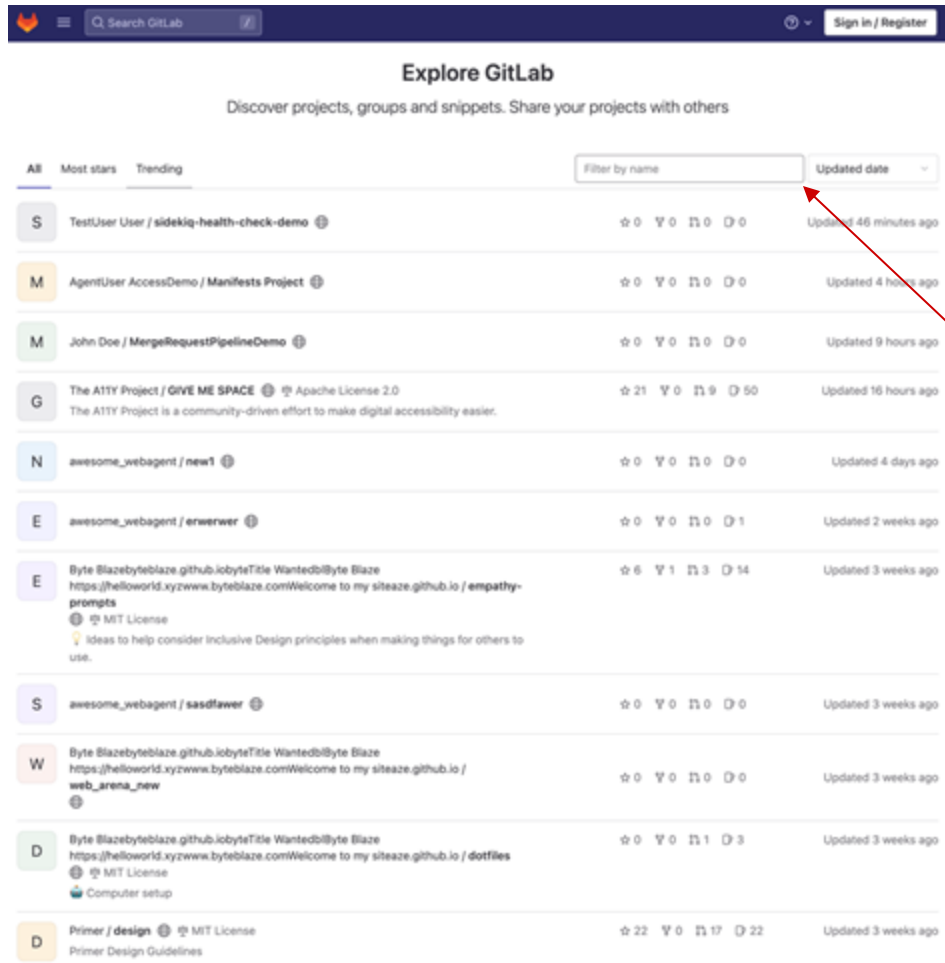
```
async def search_projects_by_keyword(keyword: str):  
    """  
    Search for projects by a specific keyword.  
    """  
  
    This function locates the 'Filter by name' search box, types the given keyword,  
    and presses 'Enter' to execute the search operation.  
  
    Usage:  
    Call this function with the desired keyword to perform a search operation  
    within the projects.  
  
    :param keyword: The keyword to search for in projects.  
    """  
  
    searchbox_locator = page.get_by_role('main', name='').get_by_role(  
        'searchbox', name='Filter by name')  
    await searchbox_locator.click()  
    await searchbox_locator.fill('')  
    await searchbox_locator.type(keyword)  
    await page.keyboard.press('Enter')
```

```
async def explore_and_clone_repository(project_name: str) -> str:  
    """  
    Explore and clone a project repository by obtaining its clone URL.  
    """  
  
    This function automates the process of navigating to the 'Explore' section of the website,  
    selecting a project from the list, accessing the project page, and retrieving the clone URL by  
    interacting with the 'Clone' options.  
  
    Usage:  
    Call this function with the 'project_name' of the project you wish to clone. It will return the  
    SSH clone URL of the repository as a string.  
  
    :param project_name: The name of the project to explore and clone.  
    :return: A string containing the SSH URL to clone the project repository.  
    """  
  
    await page.get_by_role('main', name='').get_by_role('link', name='Explore')  
    .click()  
    await page.get_by_role('main', name='heading', exact=True).get_by_role(  
        'link', name=project_name).click()  
    await page.get_by_role('main', name='').get_by_role('link', name='Clone')  
    .click()  
    await page.get_by_role('main', name='').get_by_role('button', name=  
        'Copy URL').click()  
    await page.get_by_role('main', name='').get_by_role('link', name='Clone')  
    .click()  
    return 'ssh://git@194.107.112.19:2222/(/)/git -format(%f) project_name.  
    split(' / ')[1].lower(), project_name.split(' / ')[1].replace('@', ':').  
    lower()
```

```
async def merge_project_branches(project_name: str, merge_request_title: str):  
    """  
    Explore and merge branches for a project that has open merge requests.  
    """  
  
    This function navigates to the list of open merge requests for a given project, selects a specific merge request by its title,  
    explores the merging options, and performs the merge if it is deemed acceptable (even if there are unverified changes).  
  
    Usage:  
    Call this function with the 'project_name' to open the merge request list, and 'merge_request_title' to select and  
    merge the branches.  
  
    :param project_name: The name of the project which contains the merge requests.  
    :param merge_request_title: The title of the merge request to explore and potentially merge.  
    """  
  
    await page.get_by_role('main', name='').get_by_role('link', name='2')  
    .click()  
    await page.get_by_role('main', name='').get_by_role('link', name=  
        merge_request_title).click()  
    await page.get_by_role('main', name='').get_by_role('group', name='')  
    .get_by_role('button', name='Merge...').click()  
    await page.get_by_role('dialog', name='Merge unverified changes?')  
    .get_by_role('contentinfo', name='').get_by_role('button', name=  
        'Merge unverified changes').click()  
    return 'The merge has been completed successfully.'
```

# More Examples of Synthesized APIs

## APIs in GitLab



```
async def search_projects_by_keyword(keyword: str):
```

```
    """
```

```
    Search for projects by a specific keyword.
```

```
    This function locates the 'Filter by name' search box, types the given keyword,
    and presses 'Enter' to execute the search operation.
```

```
    Usage:
```

```
    Call this function with the desired keyword to perform a search operation
    within the projects.
```

```
    :param keyword: The keyword to search for in projects.
```

```
    """
```

```
    searchbox_locator = page.get_by_role('main', name='').get_by_role('
```

```
        'searchbox', name='Filter by name')
```

```
    await searchbox_locator.click()
```

```
    await searchbox_locator.fill('')
```

```
    await searchbox_locator.type(keyword)
```

```
    await page.keyboard.press('Enter')
```

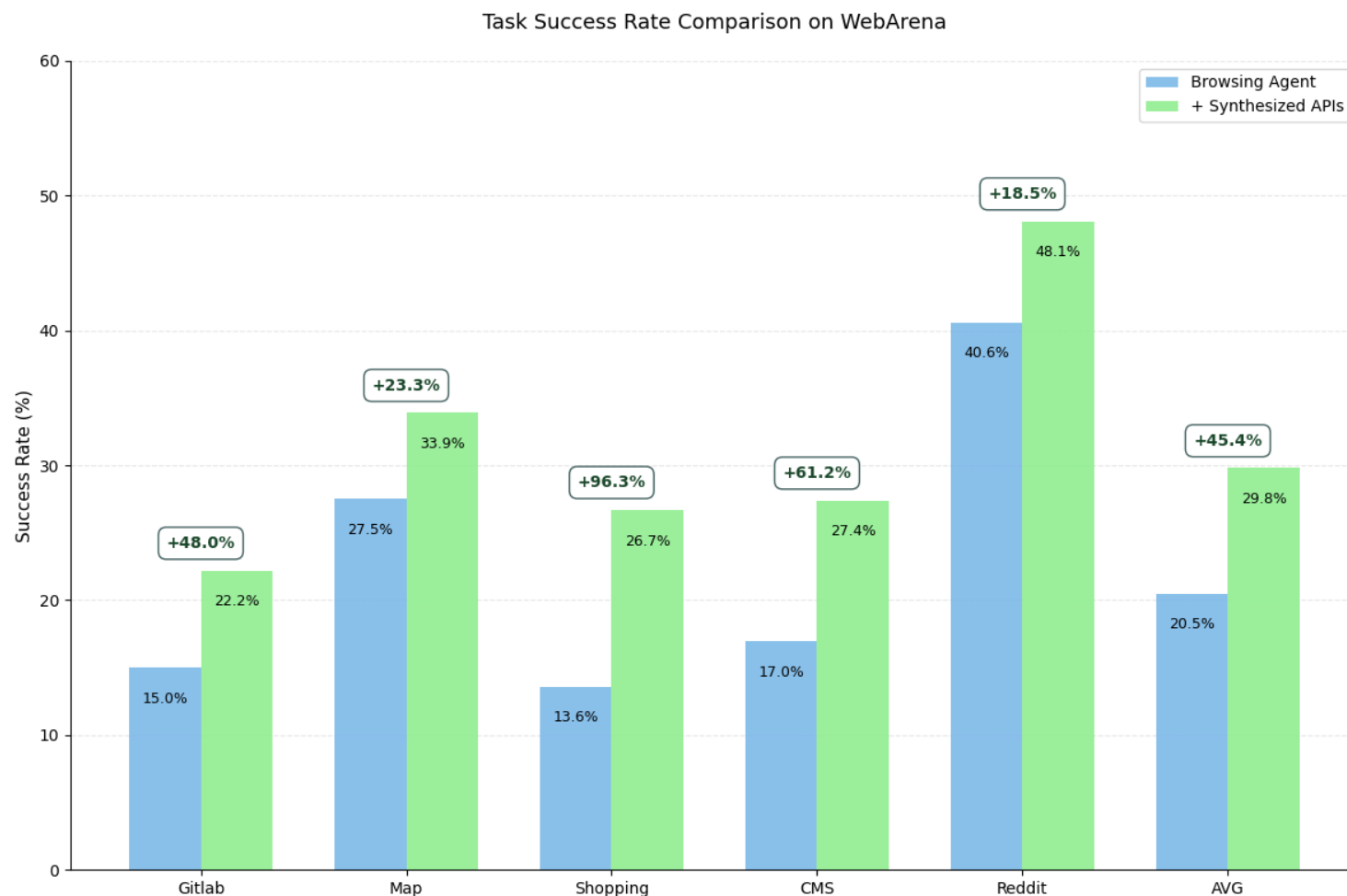
# Evaluation on Sandbox

## Exploration:

Synthesize APIs with 160 Iterations

## Agents:

- Browsing Agent:
  - Browser Action
- +API Synthesis
  - Extend action space with APIs



# How far are we from human-crafted APIs?

- Human Crafted APIs:
  - Mining APIs from sandbox source code

Not realistic, but can serve for a case study

API Documentation	<pre># Commits ## GET /api/{id}/commits: Get a list of commits in a project.   Attribute   Type             Description     `id`        integer/string   The ID or path of the project.     `since`     string           Only commits after or on this date.     `until`     string           Only commits before or on this date.   Output: JSON containing all commits that meet the given criteria.</pre>
API Calling	<pre>&lt;execute_ipython&gt; requests.get('gitlab.com/api/allyproject/commits') &lt;/execute_ipython&gt;</pre>
JSON Output	<pre>[ .....{   "id": "ed37a2f2",   "created_at": "2023-03-13T21:04:49.000-04:00",   "title": "Update README.md",   "message": "Update README.md",   "author": "SaptakS", }]</pre>



# Comparing with human crafted APIs

## API Support Level

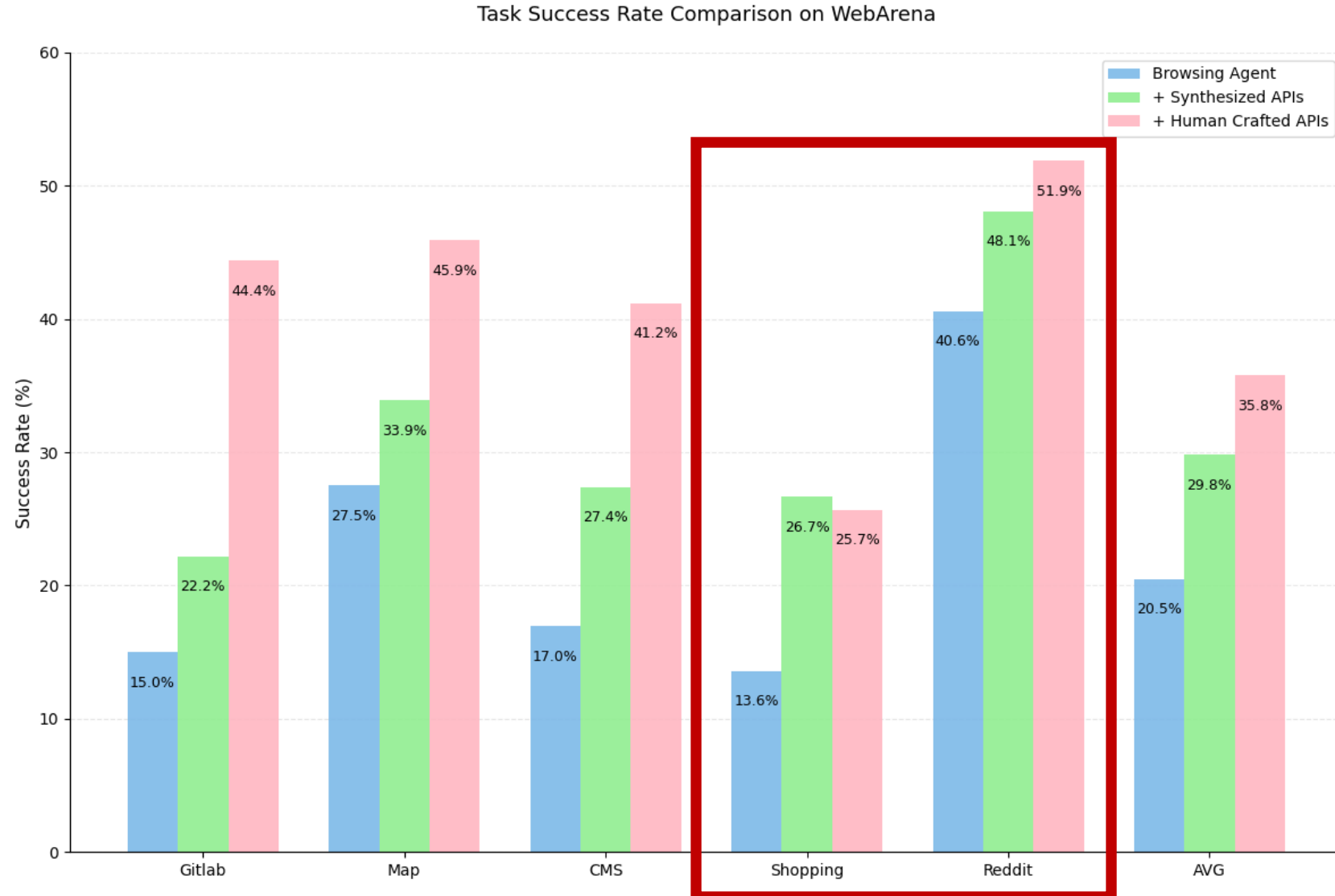
### MEDIUM:

Admin Access

Limited APIs

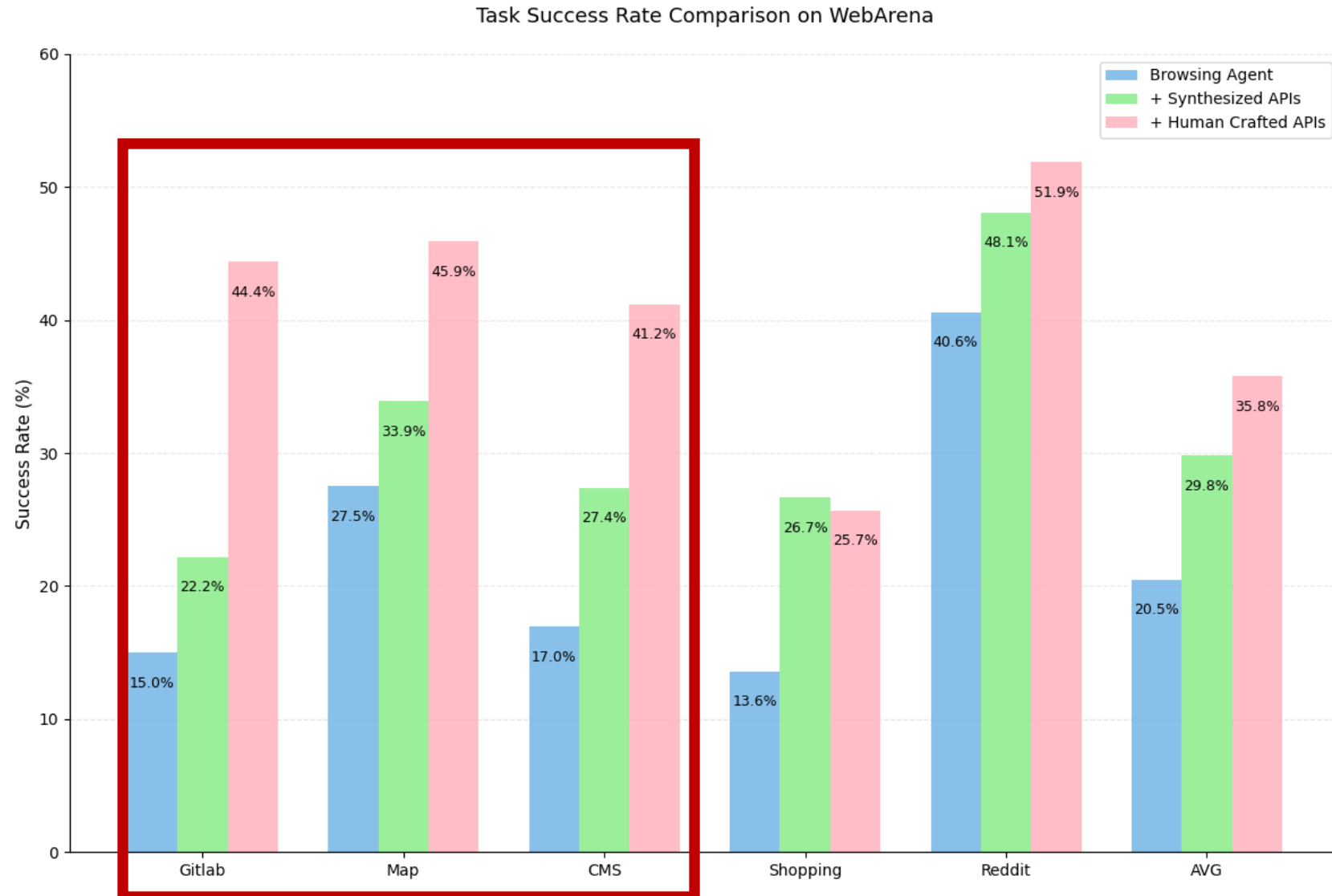
### LOW:

Manually written  
automation Code



# Comparing with human crafted APIs

**HIGH** API support:  
Admin Access  
High Quality API



# Generalization to weaker Agent

- Empowering smaller models with APIs synthesized by larger models
- Exploration: GPT-4o
- Inference: GPT-4o-mini

SKILLWEAVER						
GPT-4o	17.8	27.5	19.8	18.7	37.7	22.6
+ Skills	22.2	33.9	27.2	25.8	50.0	29.8
$\Delta$	↑ 25%	↑ 23%	↑ 38%	↑ 38%	↑ 33%	↑ 32%
GPT-4o-mini	6.1	10.3	11.8	3.3	18.9	9.2
+ Skills	8.9	16.7	17.1	7.7	26.4	14.1
$\Delta$	↑ 46%	↑ 62%	↑ 46%	↑ 133%	↑ 40%	↑ 45%

# Scaling to Live Websites

Exploration:

80 Iterations

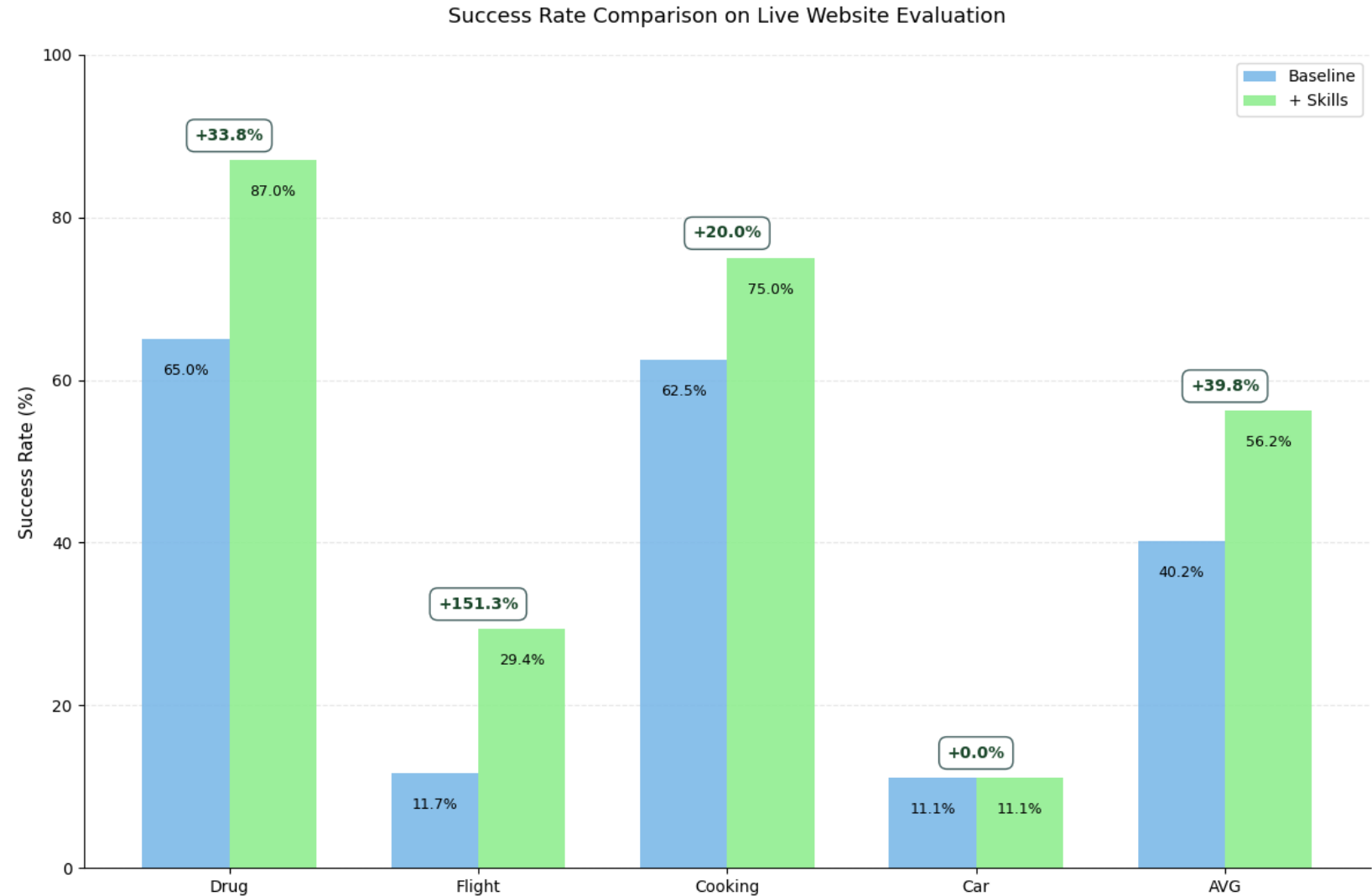
Evaluation:

Drug: 23

Flight: 17

Cooking: 8

Car: 9



*Thank you!*  
*&*  
*Questions?*