

Planning



Planning: (simplified) definition

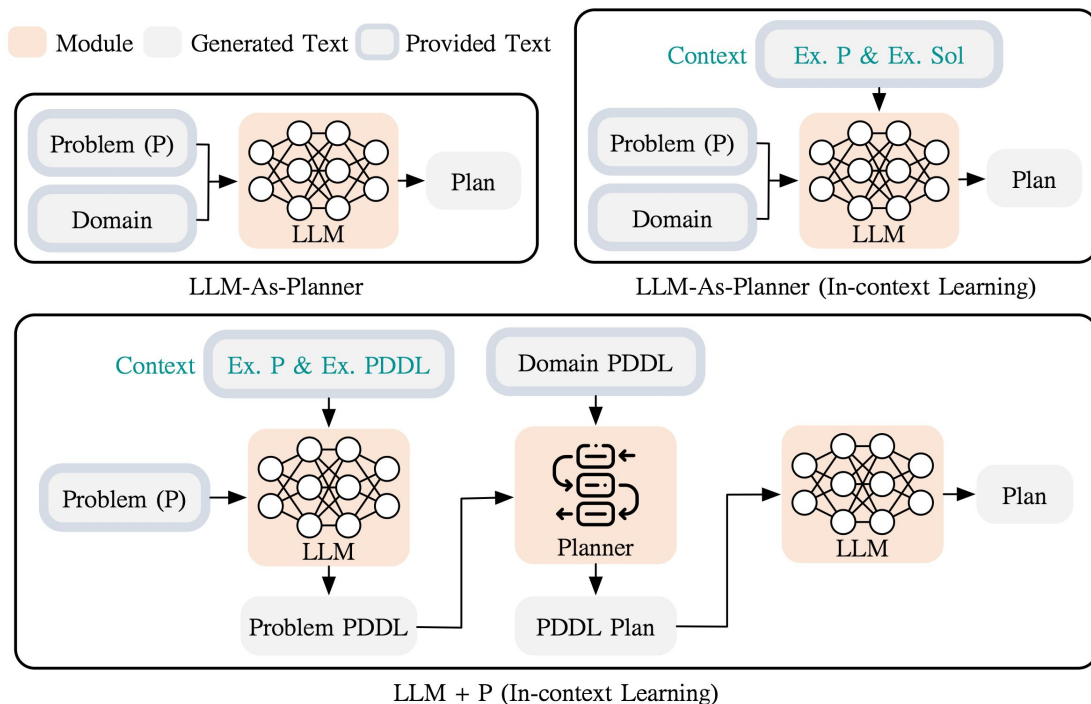
Given a goal G , decides on a sequence of actions (a_0, a_1, \dots, a_n) that will lead to a state that passes the goal test $g(\cdot)$

General trends in planning settings for language agents

- Increasing expressiveness in **goal specification**, e.g., in natural language as opposed to formal language
- Substantially expanded or open-ended **action space**
- Increasing difficulty in automated **goal test**

LLM+P

When domain and problem can be (fully or partially) described formally, e.g., in PDDL (Planning Domain Definition Language)



An Example PDDL Problem File Written by GPT-4 with Context

Context (shaded):

An example planning problem is: You have 5 blocks. b2 is on top of b5. b5 is on top of b1. b1 is on top of b4. b3 is on top of b2. b4 is on the table. b3 is clear. Your arm is empty. Your goal is to move the blocks. b4 should be on top of b3. The problem PDDL is:

```
(:objects b1 b2 b3 b4 b5)
(:init (arm-empty) (on b1 b4) (on
b2 b5) (on b3 b2)
(on-table b4) (on b5 b1)
(clear b3))
(:goal (and (on b4 b3))))
```

Prompt: Description of Problem (P1) + Context + Provide me with the problem PDDL file that describes the planning problem directly without further explanations.

GPT-4 (the generated problem PDDL):

```
(:objects b1 b2 b3 b4 b5 )
(:init (arm-empty) (on-table b1)
(on b2 b1) (on b3 b4)
(on b4 b2) (on b5 b3) (clear
b5)))
(:goal (and (on b1 b2) (on b3 b5)
(on b4 b1)))
```

Planner:

```
(unstack b5 b3) (putdown b5)
(unstack b3 b4) (stack b3 b5)
(unstack b4 b2) (putdown b4)
(unstack b2 b1) (putdown b2)
(pickup b1) (stack b1 b2) (pickup
b4) (stack b4 b1)
```

Language agent planning: web agents

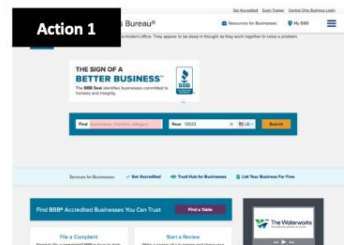
Task Description:

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

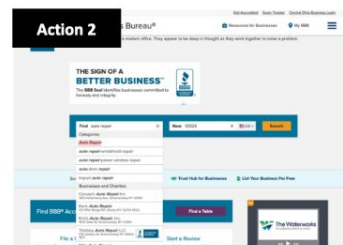
Action Sequence:

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

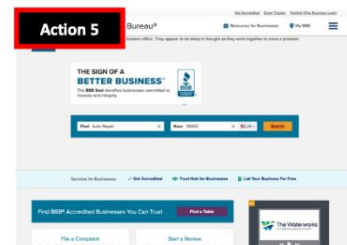
Webpage Snapshots:



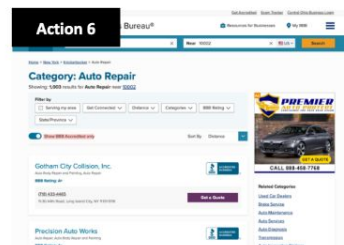
`<input name="find_text" type="search">`



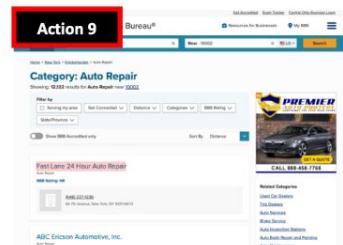
`Auto Repair`



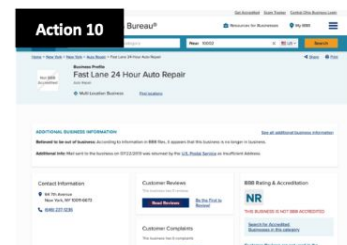
`<button>Search</button>`



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

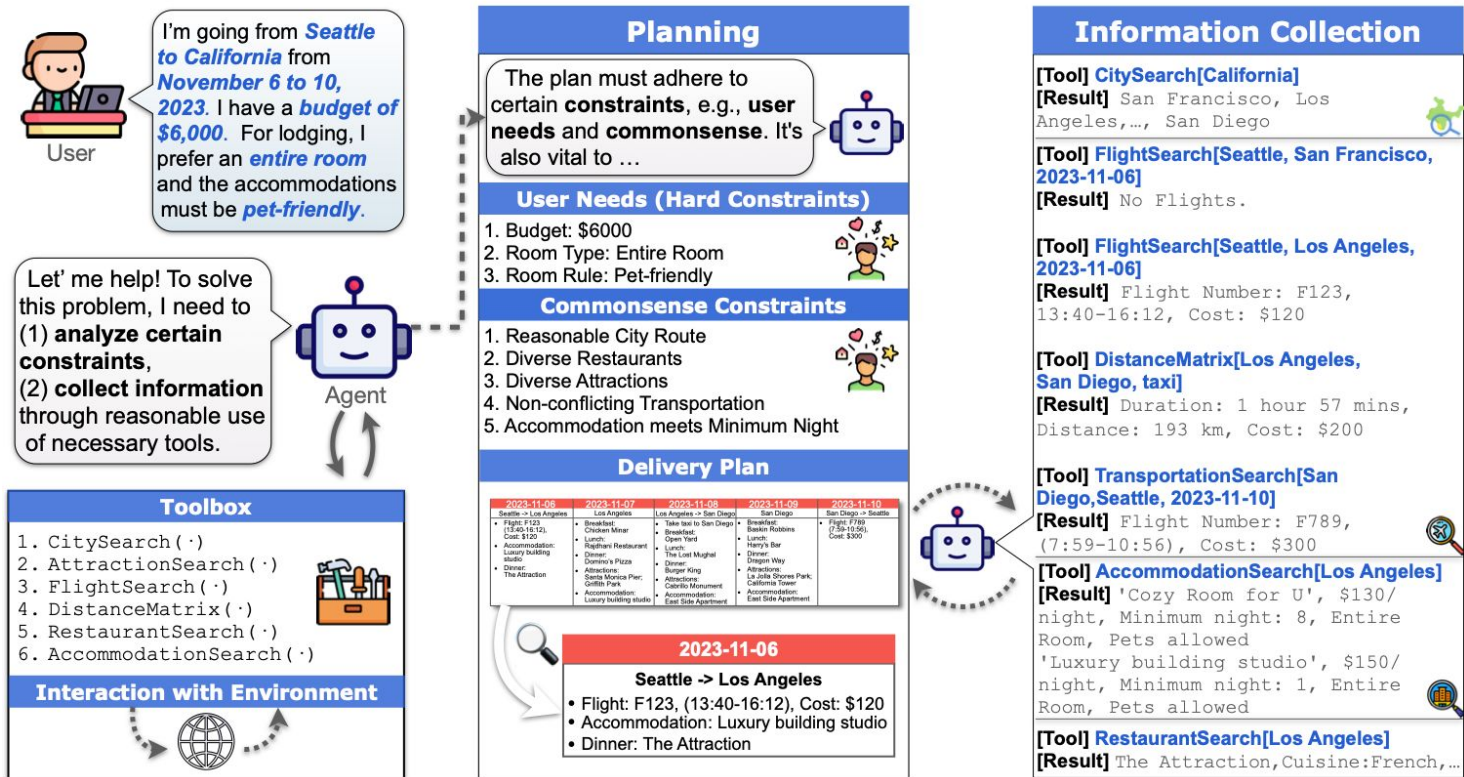


`Fast Lane 24 Hour Auto Repair`



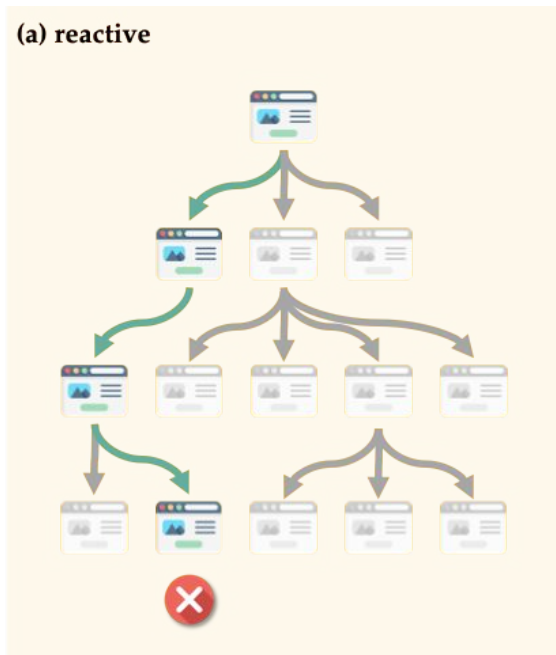
`Read Reviews`

Language agent planning: travel planning

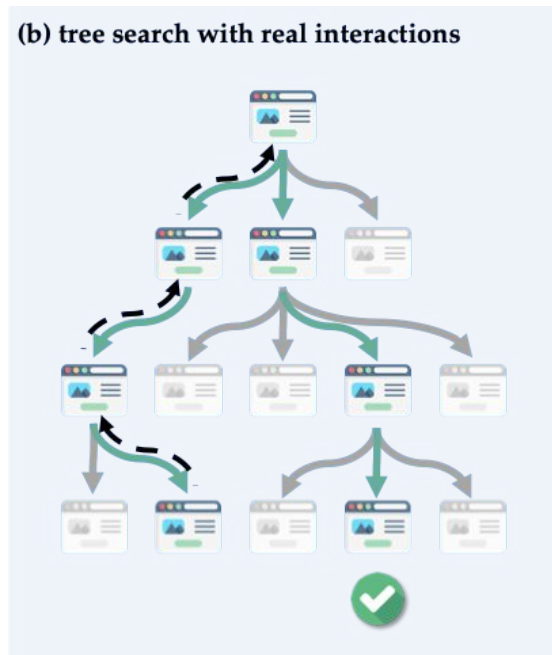


Planning paradigms for language agents

(a) reactive



(b) tree search with real interactions

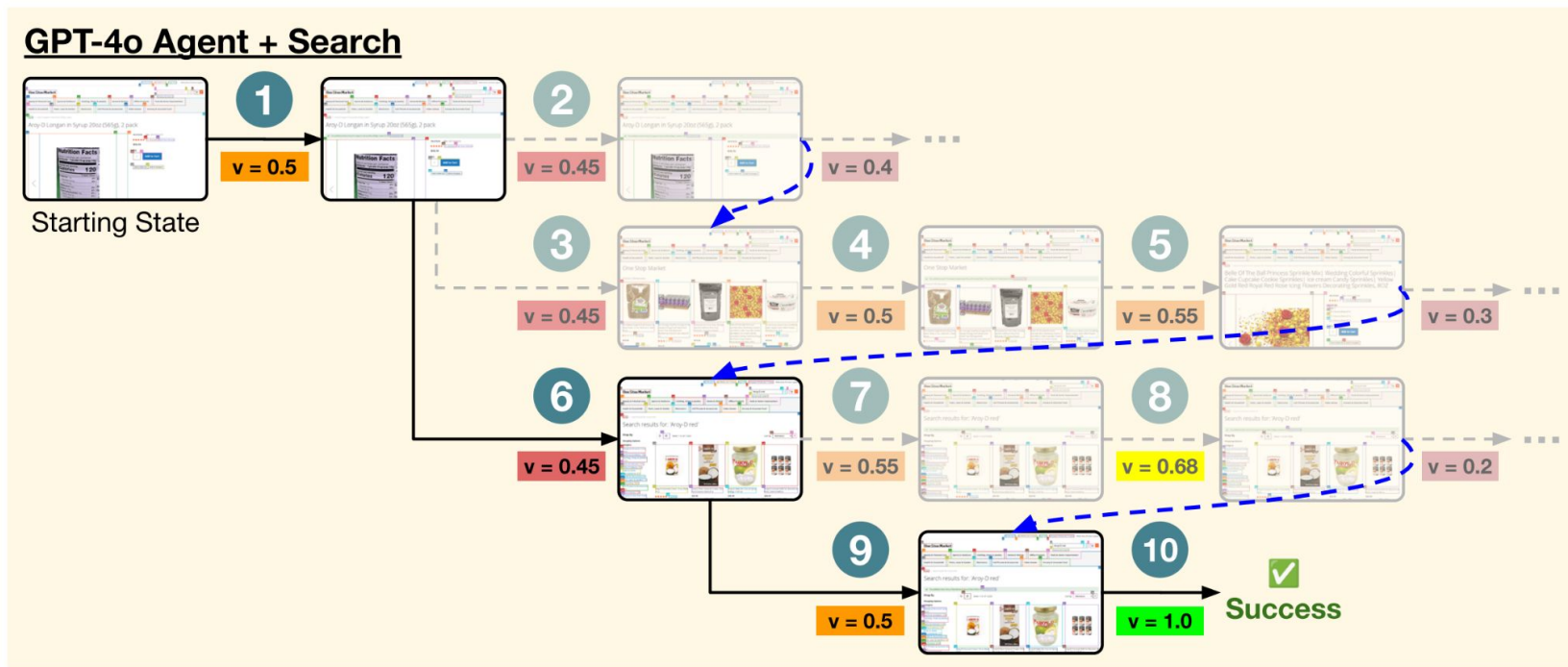


fast, easy to implement



greedy,
short-sighted

Tree search with real interactions




Jing Yu Koh, Stephen McAleer, Daniel Fried, Ruslan Salakhutdinov. "Tree Search for Language Model Agents." arXiv preprint arXiv:2407.01476 (2024).
Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, Yu-Xiong Wang. "Language Agent Tree Search Unifies Reasoning Acting and Planning in Language Models." ICML (2024).

Challenges with tree search in the real world

- Many actions are state-changing and irreversible → backtracking ❌
- Safety/privacy risks
- Inference-time exploration could be slow and costly

Cancel Place Your Order - Amazon.co...

 Nespresso Capsules Vertuo, Variety Pack, Medium and Dark Roast Coffee, 30 Count Coffee Pods, Brews 7.8 oz.
\$37.50 (\$1.25 / Count)
✓prime
Ships from and sold by Amazon.com

Quantity: 1 Change
Add gift options
Auto-deliver and save up to 5% on future auto-deliveries
Item often ships in manufacturer's container to reduce packaging and reveals what's inside. To change, click below.

Reduce packaging, ship in manufacturer's container ▼

Place your order

By placing your order, you agree to Amazon's [privacy notice](#) and [conditions of use](#).

← Search or ask a question

Kohl's Dropoff FREE

Kohl's will pack, label, and ship your return for free. Just bring the item in its original manufacturer's packaging and disassemble the item (if applicable). We'll email you a QR code to ship your return. Show it to a store associate at any Kohl's store.

[Find a participating Kohl's store](#)
Printer not required.

☐ The UPS Store locations only—no label needed \$6.99

☐ Amazon Dropoff — box and label needed FREE

2 OTHER RETURN OPTIONS ▼

Refund summary \$13.21 ▼

Confirm your return

Verify mobile number

A text with a One Time Password (OTP) has been sent to your mobile number: **8058671234** [Change](#)

Enter OTP: [Resend code](#)

Create your Amazon account

By creating an account, you agree to Amazon's [Conditions of Use](#) and [Privacy Notice](#).

← Search or ask a question

Location

Disabled ▼

AddressBook/Checkout

Your current location will be used to assist in adding a new address to your Amazon address book.

Amazon Cash

We use your location to find nearby stores where you can add money to your Amazon balance with Amazon Cash.

Branded Store Experience Location

We use your location to power branded store experiences.

Branded Store Experience Location-based Augmented Reality

We use your camera, motion, and location to power branded store experiences. Requires camera, motion, and location.

Campus pickup

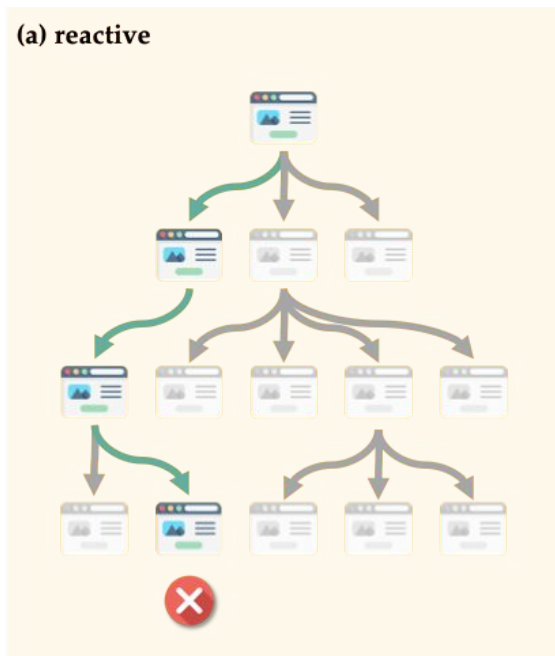
We'll use your location to show the nearby pickup points



Delivery Location

We use your location to improve your shopping experience, ensuring you only see products and delivery options available in your area.

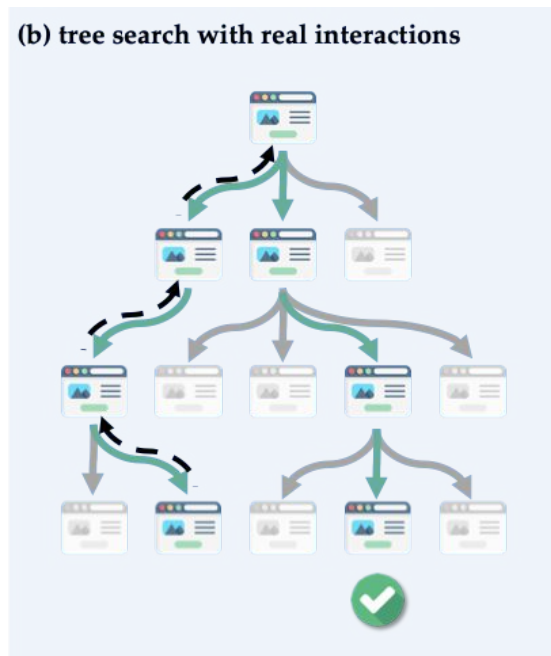
Planning paradigms for language agents



(a) reactive



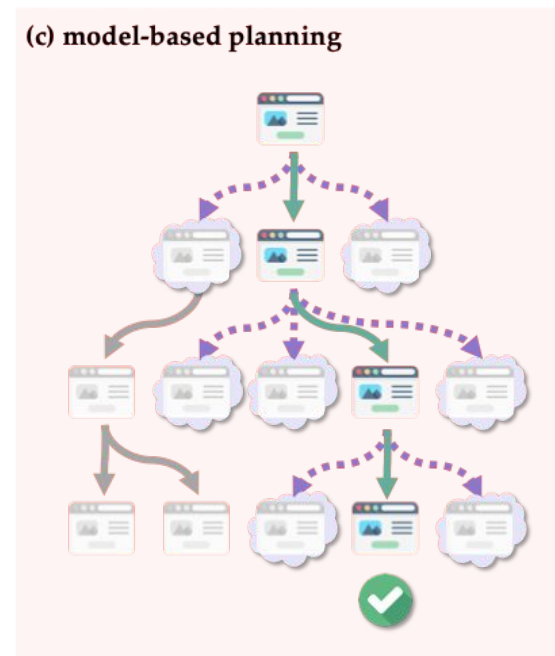
-  fast, easy to implement
-  greedy, short-sighted



(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?

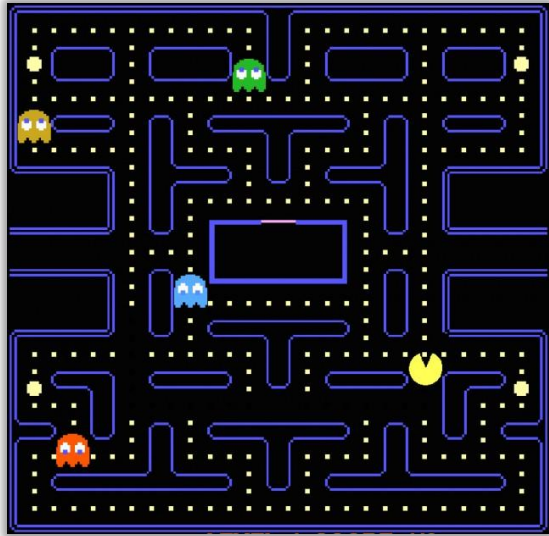
What's ... a world model?

A computational model of environment transition dynamics

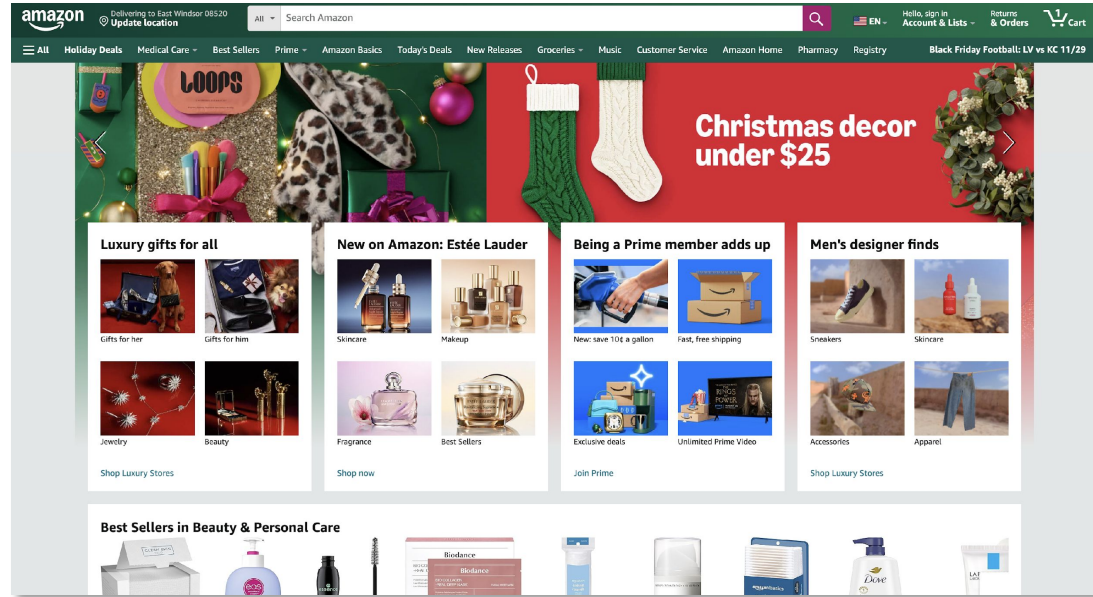
$$\hat{T}: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$$

If I do this (\mathbf{a}_t) right now (\mathbf{s}_t), what would happen next (\mathbf{s}_{t+1})?

Why hasn't it been done yet?

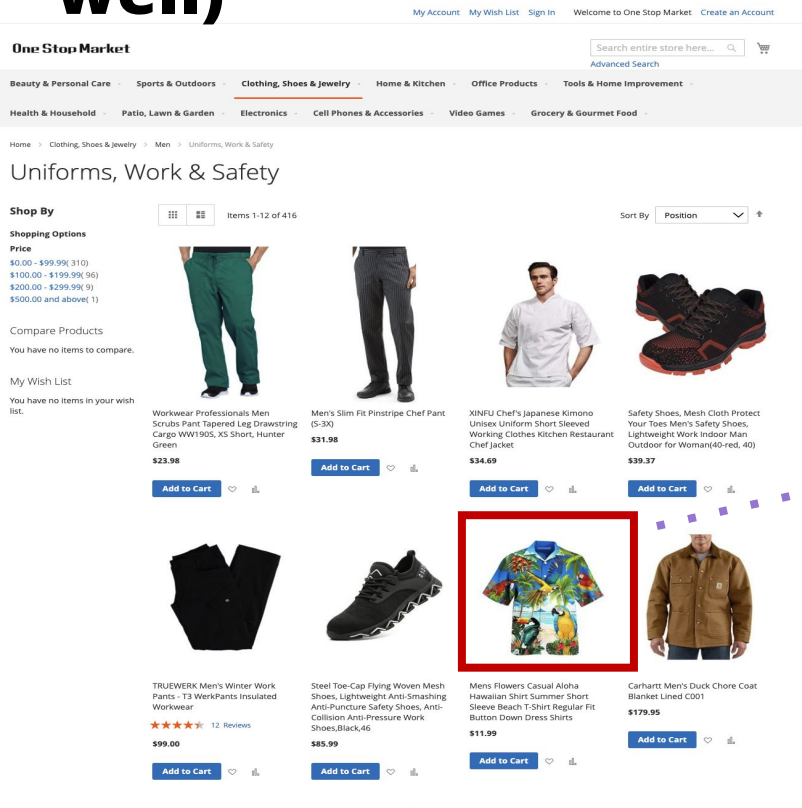


V.S



And **billions of other websites** on the Internet!

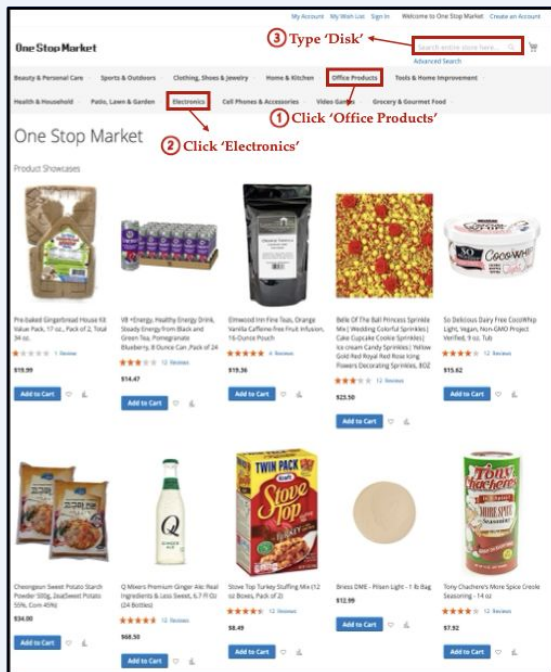
LLMs can predict state transitions (reasonably well)



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.

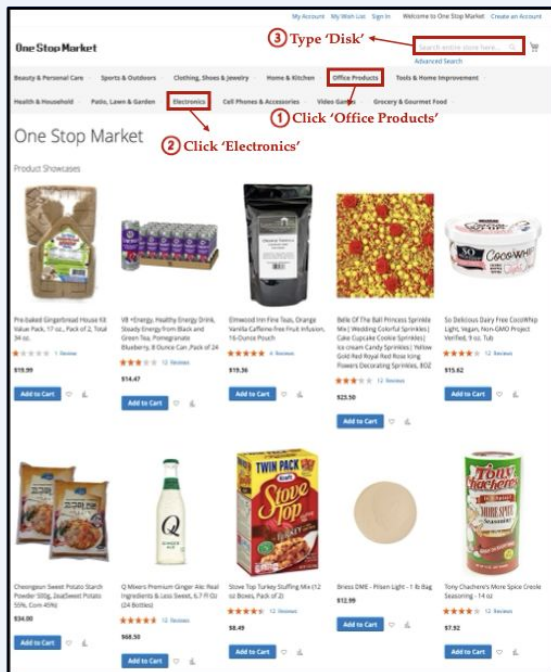
WebDreamer: model-based planner

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



WebDreamer: model-based planner

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



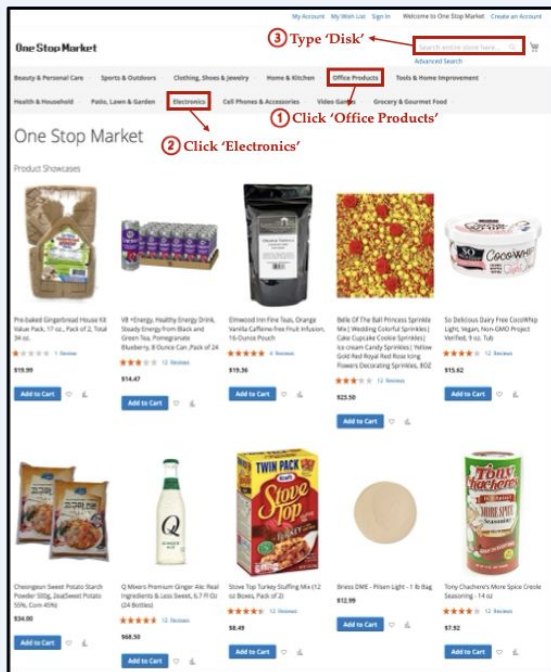
Stage I: Simulation

1

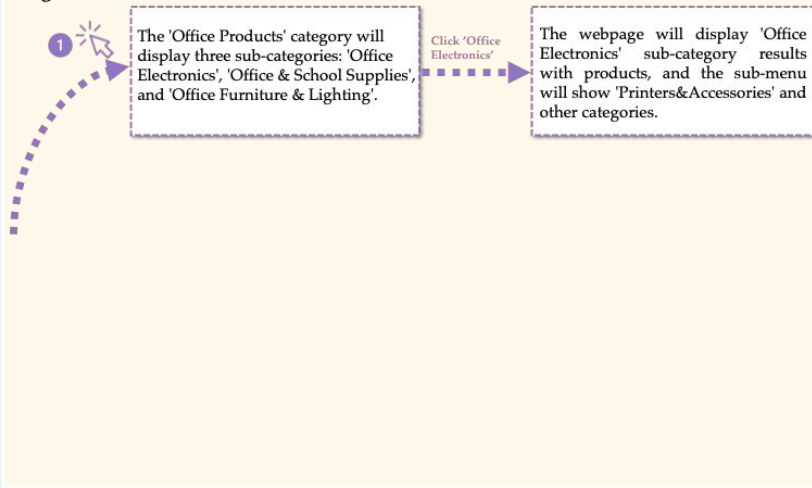
The 'Office Products' category will display three sub-categories: 'Office Electronics', 'Office & School Supplies', and 'Office Furniture & Lighting'.

WebDreamer: model-based planner

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

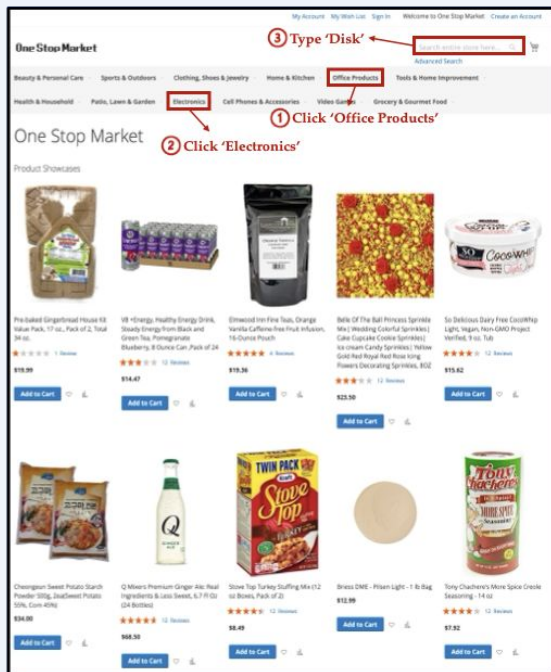


Stage I: Simulation

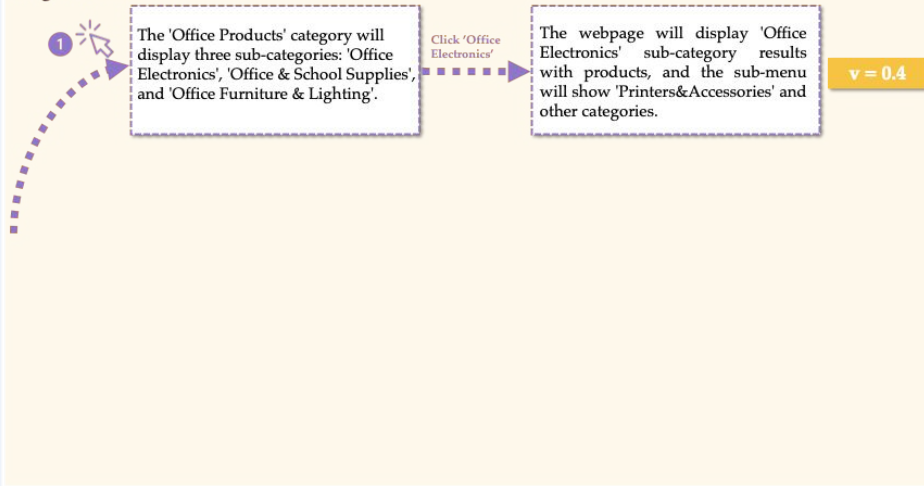


WebDreamer: model-based planner

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

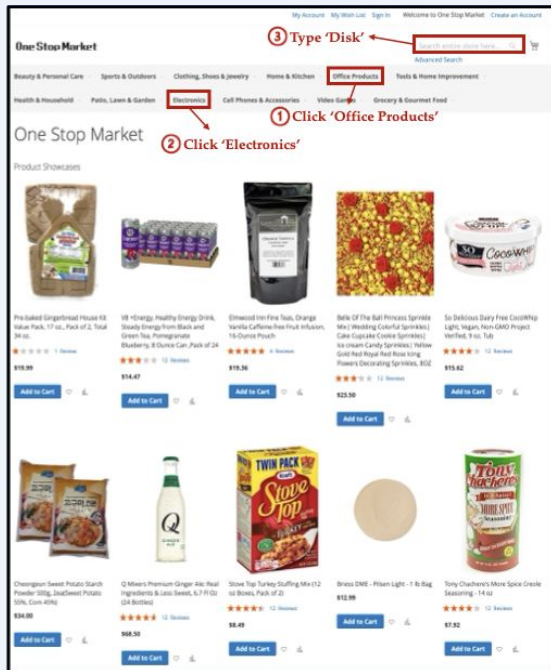


Stage I: Simulation

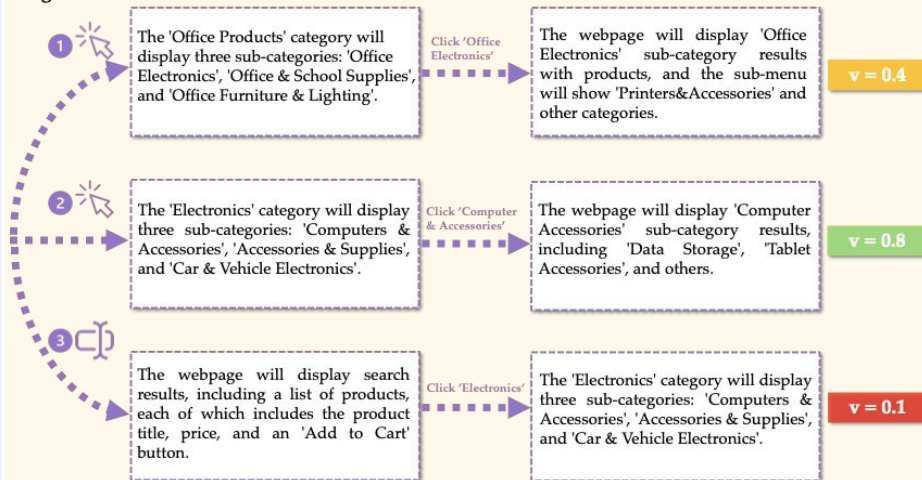


WebDreamer: model-based planner

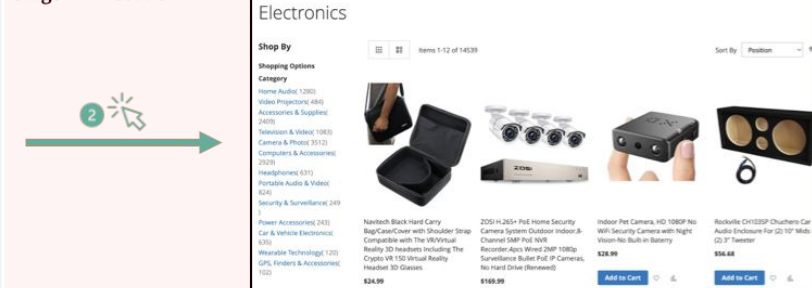
Please navigate to the 'Data Storage' category and purchase the least expensive disk with 512GB of storage.



Stage I: Simulation

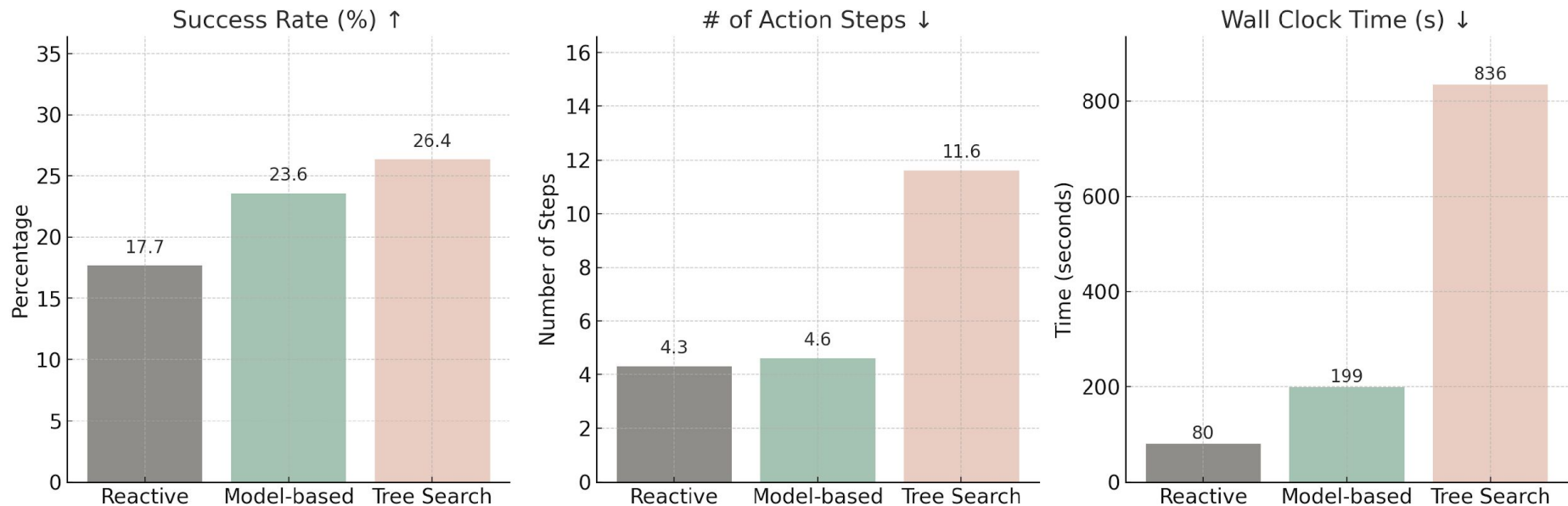


Stage II: Execution



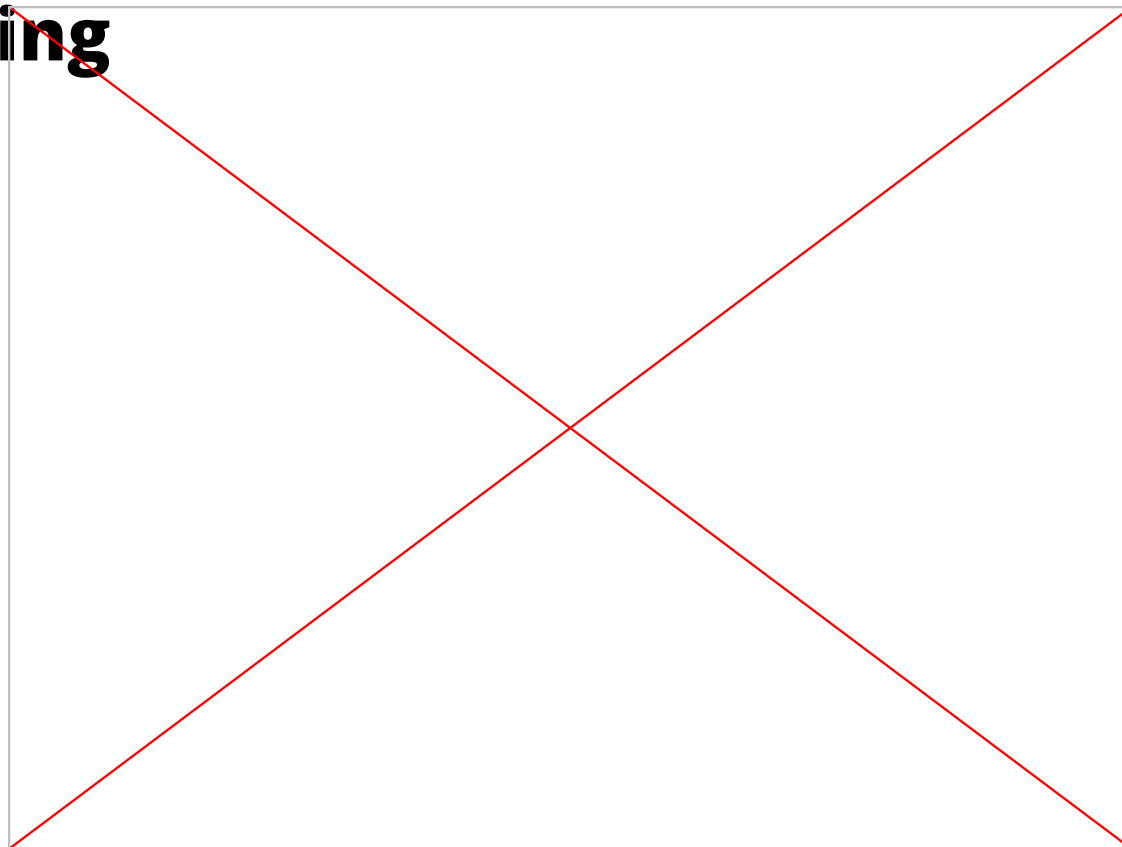
Results on VisualWebArena

Model-based planning is more accurate than reactive planning and more efficient than tree search (also recall the challenges of tree search in real-world environments)



Reactive and tree search results are from Koh et al., (2024)

Hierarchical planning and dynamic replanning



Hierarchical planning and dynamic rep

The screenshot displays the Devin AI interface within a web browser. The browser tab is labeled 'Devin (the developer)' and the address bar shows 'preview.devin.ai'. The interface is divided into two main sections: 'Benchmark Llama 2 Providers' on the left and 'Devin's Workspace' on the right.

Benchmark Llama 2 Providers

Scott
Hey Devin, I'd like for you to benchmark Llama 2 on three different providers: Replicate, Together, and Perplexity. Figure out their API formats and write a script that sends the same prompt/params to all of them

Devin
Got it, I'll look into the API formats for Replicate, Together, and Perplexity and create a script to benchmark Llama 2 across these services. I'll get started on that now.

Devin is researching the API documentation for Replicate, Together, and Perplexity to understand their request formats and parameters.

Send a message (won't interrupt Devin)

Devin's Workspace

Shell Browser Editor **Planner**

Current Focus: Research the API documentation for Replicate, Together, and Perplexity.

Plan:

- ☐ Research the API documentation for Replicate, Together, and Perplexity to understand their request formats and parameters.
- ☐ Write a Python script that constructs and sends HTTP requests to each API with the same prompt and parameters.
- ☐ Implement response time measurement and accuracy assessment within the script to benchmark each API's performance.
- ☐ Test the script to ensure it works correctly and gathers the necessary data from each API.
- ☐ Analyze the collected data to compare the performance of the three APIs.
- ☐ Document the script and the benchmarking process, including any dependencies or environment setup needed.
- ☐ Report the findings back to the user with the collected data and any relevant visualizations or summaries.

At the bottom of the workspace, there are icons for a list, previous, next, and a 'Live' indicator.

High-level
Plan

Hierarchical planning and dynamic rep

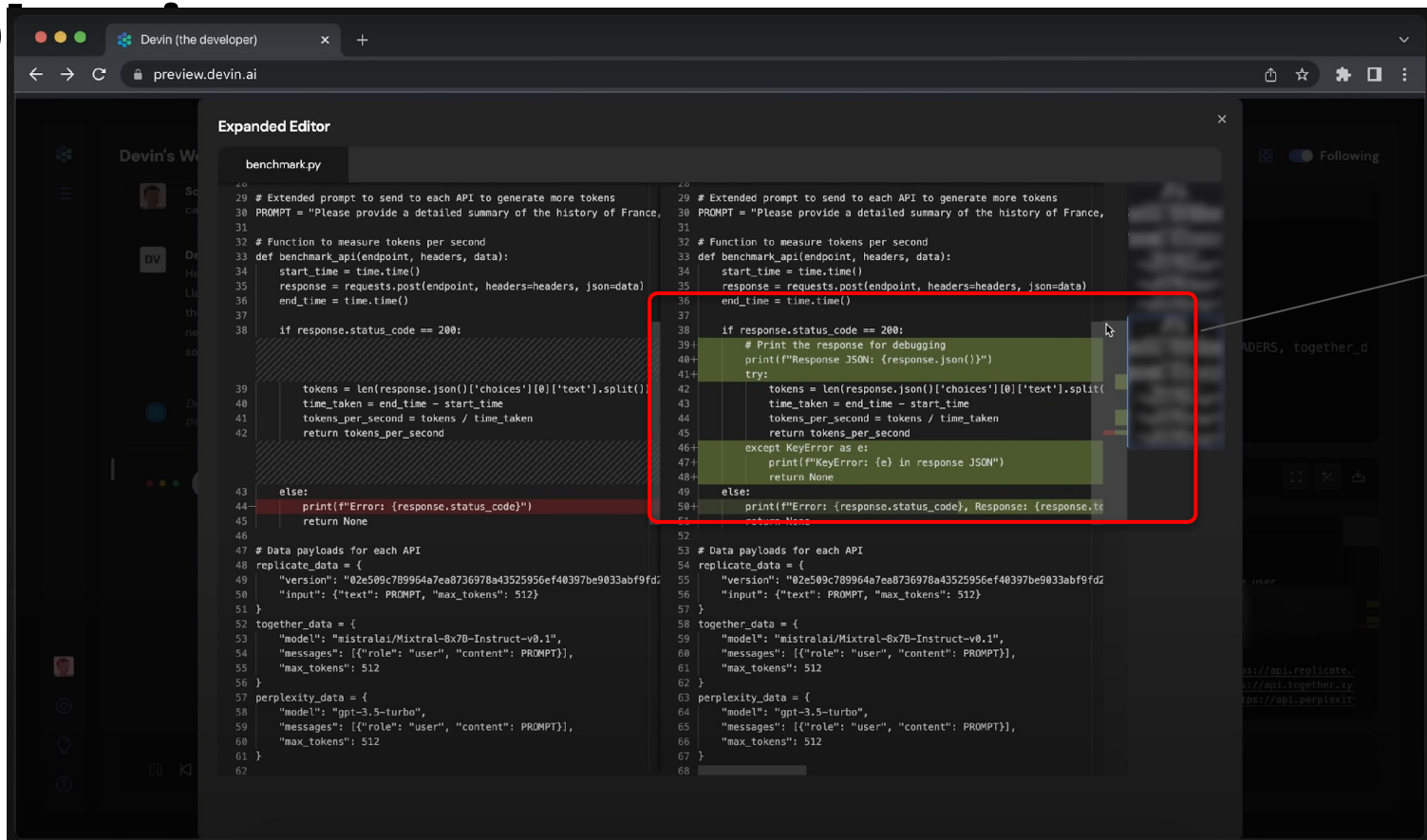
The screenshot shows the Devin AI interface. On the left, a chat window titled "Devin's Workspace" shows a conversation between Scott and Devin. Devin is currently debugging a script. On the right, a code editor shows the "benchmark.py" file. A red box highlights an error message in the terminal:

```
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
ubuntu@ip-172-31-27-196:~$ python3 benchmark.py
Error: 422
Traceback (most recent call last):
  File "/home/ubuntu/benchmark.py", line 65, in <module>
    together_tps = benchmark_api(TOGETHER_ENDPOINT, TOGETHER_HEADERS, together_d
ata)
  File "/home/ubuntu/benchmark.py", line 39, in benchmark_api
    tokens = len(response.json()['choices'][0]['text'].split())
KeyError: 'text'
```

An arrow points from the text "Ran into exception when carrying out a subgoal" to the error message. Below the chat, a browser window shows the "together.ai" documentation for "Chat Completions".

Ran into exception when carrying out a subgoal

Hierarchical planning and dynamic rep

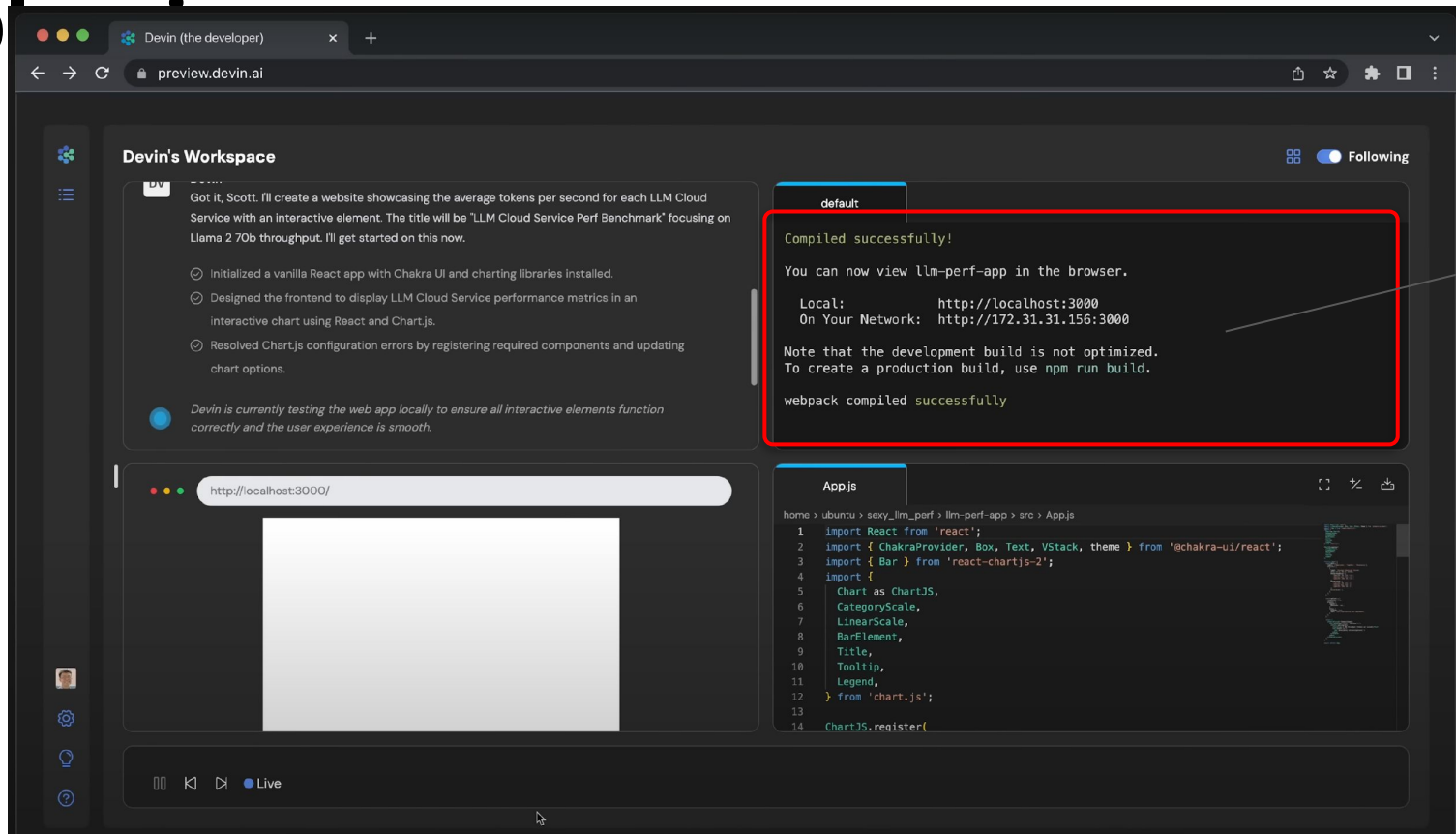


The screenshot displays the 'Expanded Editor' of the Devin AI interface, showing a side-by-side comparison of a Python file named 'benchmark.py'. The interface includes a browser-like header with navigation icons and a sidebar on the left showing project files. The code is split into two panels, each showing a different version of the script. A red rectangular box highlights a specific section of the code in the right-hand panel, which is an exception handling block. An arrow points from the text 'Re-plan and add print() around the exception to get debug info' to this highlighted section.

```
29 # Extended prompt to send to each API to generate more tokens
30 PROMPT = "Please provide a detailed summary of the history of France,
31
32 # Function to measure tokens per second
33 def benchmark_api(endpoint, headers, data):
34     start_time = time.time()
35     response = requests.post(endpoint, headers=headers, json=data)
36     end_time = time.time()
37
38     if response.status_code == 200:
39
40         tokens = len(response.json()['choices'][0]['text'].split())
41         time_taken = end_time - start_time
42         tokens_per_second = tokens / time_taken
43         return tokens_per_second
44
45     else:
46         print(f"Error: {response.status_code}")
47         return None
48
49 # Data payloads for each API
50 replicate_data = {
51     "version": "02e509c789964a7ea8736978a43525956ef40397be9033abf9fd",
52     "input": {"text": PROMPT, "max_tokens": 512}
53 }
54
55 together_data = {
56     "model": "mistralai/Mistral-8x7B-Instruct-v0.1",
57     "messages": [{"role": "user", "content": PROMPT}],
58     "max_tokens": 512
59 }
60
61 perplexity_data = {
62     "model": "gpt-3.5-turbo",
63     "messages": [{"role": "user", "content": PROMPT}],
64     "max_tokens": 512
65 }
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Re-plan
and add
print()
around the
exception
to get
debug info

Hierarchical planning and dynamic rep



Solved the exception based on the printed out debug info!

Planning: Takeaways

- Language agents are expanding into new planning scenarios
 - characterized by expressive but fuzzy goal specifications, open-ended action spaces, more difficult and sometimes non-binary goal tests
- Language for reasoning also enables new planning abilities
 - Generalist world models and model-based planning
 - Hierarchical planning and dynamic replanning
- The best planning strategy is dependent on the LLM; stronger LLMs may require less scaffolding (i.e., more ‘reactive’)
- How to improve planning in LLMs is still largely an open question