On the Fine-Grained Planning Abilities of VLM Web Agents

Surgan Jandial*, **Yinong Oliver Wang***, **Andrea Bajcsy**, **Fernando De la Torre**
Carnegie Mellon University

Abstract

Vision-Language Models (VLMs) have shown promise as web agents, yet their planning—the ability to devise strategies or action sequences to complete tasks—remains understudied. While prior works focus on VLM's perception and overall success rates (i.e., goal completion), fine-grained investigation of their planning has been overlooked. To address this gap, we examine VLMs' capability to (1) understand temporal relationships within web contexts, and (2) assess plans of actions across diverse scenarios. We design four simple yet effective tests to delve into these nuanced aspects around planning. Our results across nineteen VLMs reveal that these models exhibit limited performance in the aforementioned skills and are not reliable to function as web agents. To facilitate future work, we release our planning evaluations and data, providing a foundation for advancing the future research in this area.

1 Introduction

Recent advancements in Vision-Language Models (VLMs) have shown great potential as web agents capable of autonomously executing user instructions, such as "Buy a flight ticket from A to B". To successfully accomplish these tasks, recent literatures (Koh et al., 2024b; Gu et al., 2024) have shown that VLMs must perform three key functions: perceive and comprehend the structure of a webpage and relevant elements (Perception), reason about task requirements and devise actionable plans (Planning), and execute appropriate actions to achieve the desired goal (Action). While VLMs have demonstrated promising potentials, their current performance remains inadequate for reliable web agents (Koh et al., 2024a; Liu et al., 2024d; Zheng et al., 2024; He et al., 2024).

Although limitations in model and algorithms pose a significant challenge, we argue that the lack

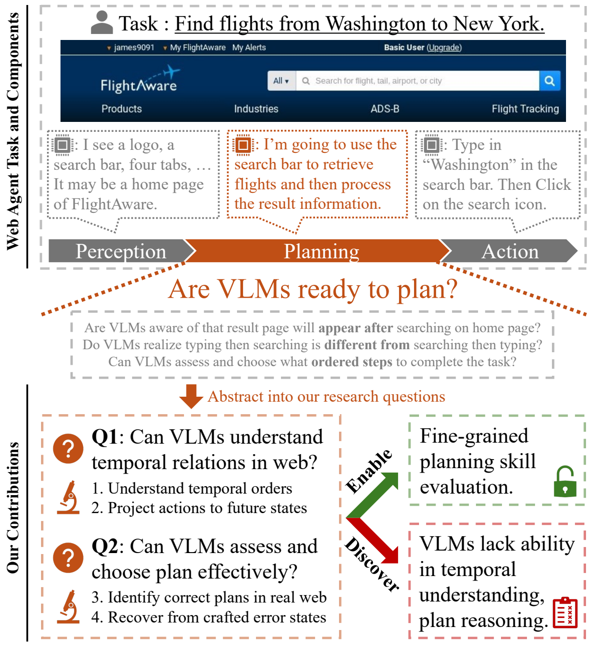


Figure 1: **VLMs and Web Planning.** For web agents to complete tasks, a set of critical components are involved. Among them, the fine-grained evaluation in the planning component is essential but overlooked. Thus, we design questions that investigate whether VLMs possess the nuanced planning skills, enabling the first web planning evaluation and concluding the lack of capabilities in VLMs to plan effectively and reliably.

of holistic evaluations is another key factor hindering progress. More precisely, prior works primarily assess the efficacy of VLM agents by judging the correctness of their actions and measuring task success rates, i.e., whether the goal is achieved (Pan et al., 2024; He et al., 2024; Deng et al., 2023). While this is intuitive and aligned with the objective of task completion, end-to-end success rates cannot reveal the bottlenecks of agents or offer detailed diagnosis and insight into the specific Perception and Planning capabilities. Observing the gap, prior works (Liu et al., 2024d; Wang et al., 2024a) prompt the evaluations of fine-grained VLM skills in web scenarios, such as website grounding and WebQA (Liu et al., 2024d; Wang et al., 2024a).

*Authors contributed Equally

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 25347–25380
November 4-9, 2025

However, they focus solely on perception skills, failing to address the underlying abilities required for effective planning as shown in Fig. 1, thus motivating our central research question: ***Have VLMs acquired the necessary capabilities to plan effectively as reliable web agents?***

To concretize our study, we define planning as an agent's ability to devise a sequence of actions to successfully complete a given task. In the context of web, a performant planning model must possess three fundamental skills. First, it needs to understand "**order**", determining which webpage states should come earlier during task completion, essentially answering the question, "What needs to be done first, and what comes later?" Second, it must grasp "**change**", recognizing how the sequence of action(s) transforms the webpage states. Third, it must be able to incorporate the order and change to "**assess**" a plan that achieves the desired goal.

Hence, we design a series of unit tests to target the above prerequisite skills through two key dimensions. In the first dimension, **Temporal Understanding** (Fig. 1 Q1), we conduct two tests in Sec. 4 to examine whether VLMs can infer which web-page state/screenshot comes earlier in the trajectory of a given task (e.g "find flights from Washington to New York"), and can predict future states following actions. Our results highlight that VLMs struggle with basic temporal understanding capabilities and they tend to rely on spurious heuristics, such as the order in which inputs are presented, instead of the actual semantic logic. Moreover, as the number of action steps increases, VLMs exhibit declining performance in predicting future states.

In the second dimension, **Plan Assessment** (Fig. 1 Q2), we examine whether VLMs can reason and choose the correct course of action to achieve a task, given a goal and a screenshot as the start state. Specifically, in Sec. 5, we test on real-world web environments and a synthetic web dataset curated for diversity and controllability. The results show overall poor performance in assessing plans, with significant struggles in handling edge cases, such as reverting erroneous actions (e.g clicking the wrong checkbox) in the synthetic dataset.

Overall, our findings reveal that VLMs are not yet equipped with adequate abilities to reliably plan as trustworthy web agents. We envision that our benchmark can pave the way for future research in thorough evaluations of VLMs and hopefully inspire ideas to advance effective and reliable web agents. In summary, our contributions are:

- We are the first to emphasize and study the fine-grained skills or prerequisites required by the VLMs to serve as reliable web agents.
- We repurpose existing datasets/environments to construct our test datasets and release for future web agent evaluations.
- We benchmark 19 state-of-the-art VLMs and conclude that current VLMs cannot be reliable web agents yet due to limitations in planning.

2 Related Work

VLMs as Web Agents. Vision Language Models (VLMs) exhibit great potential across a diverse range of challenges (Ghosh et al., 2024; Ma et al., 2024; Li et al., 2024b). Of these, one such problem is "Web Navigation" (Xu et al., 2025; Zheng et al., 2024), where given a user instruction (e.g., "Buy a pen from eBay"), and the current state (e.g., webpage screenshot, HTML or meta-data), VLMs' objective is to predict the next best action(s) that could lead to the final goal. To succeed, VLMs must excel in three key areas: 1) **Perceiving** web pages with arbitrary formats, dense images and text, 2) **Planning and Reasoning** over user's intent, extracted webpage information, and potential steps to solve the task, and 3) Predicting the next **Action** and its location on the webpage. This led to diverse benchmarks to evaluate VLMs as shown in Fig. 2.

VLM Agent Task Evaluations. Earlier benchmarks, such as MiniWob (Shi et al., 2017) and WebShop (Yao et al., preprint) typically evaluated web navigation in simple, simulated environments. More recently, works like VisualWebArena (Koh et al., 2024a), SeeAct (Zheng et al., 2024), WebVoyager (He et al., 2024), and MMInA (Zhang et al., 2024) introduced more challenging settings, testing agents on the real web. Building on this, (Shlomov et al., 2024) decomposed the agent's performance into grounding and planning by measuring the success rate on samples where VLM achieved perfect grounding. However, these benchmarks primarily assess task success, i.e., whether the final objective is achieved, in two main approaches: the first employs step-wise evaluation using human-annotated datasets, specifying the ground-truth action at each step (Deng et al., 2023; Zheng et al., 2024), whereas the second leverages contemporary VLMs (such as GPT-4o, Gemini) to determine if the agent successfully completed the task (Pan et al., 2024; He et al., 2024). Despite their intuitive nature, task success metrics have two key limita-

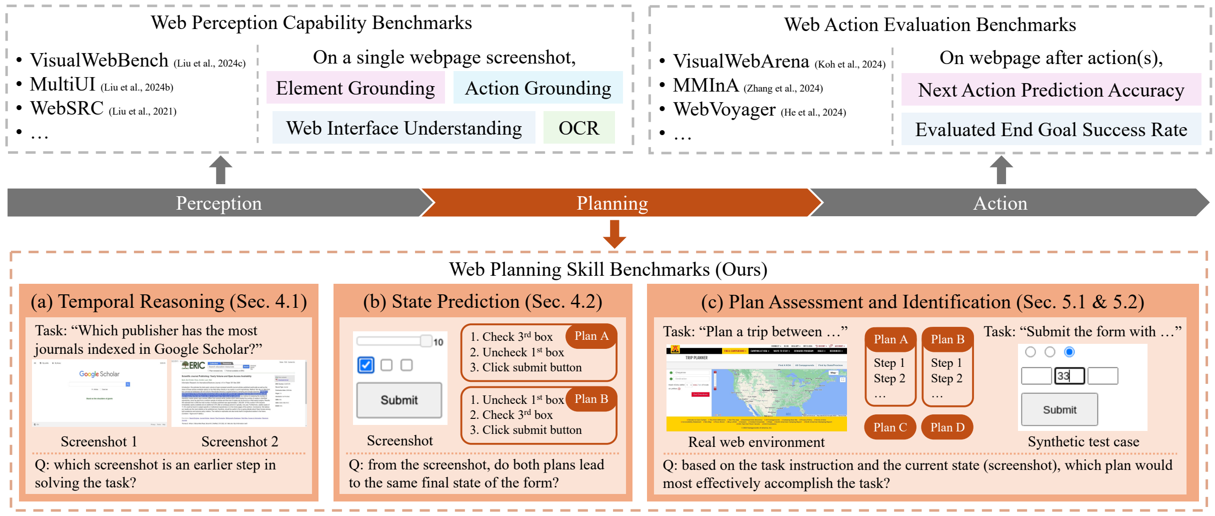


Figure 2: **Comparisons of benchmarks in different components.** While extensive study has focused on evaluating the required capabilities of the perception and action stage of web agents, the necessary skills for web planning are overlooked in existing literature and we contribute to cover this gap through four evaluations in this paper.

tions: 1). **Reliability** – As noted in (He et al., 2024), there exist multiple valid trajectories to solve a task and enforcing the agent to match the predefined action for each step makes the evaluation overly restrictive. Moreover, relying on existing VLMs/LLMs to assess task completion introduces errors due to their limited capabilities in web environments (Pan et al., 2024); 2). **Depth of Evaluation** – While task success rates measure VLMs' overall effectiveness, they lack fine-grained insights into their capabilities and defects.

VLM Web Understanding and Reasoning Evaluation. To circumvent the above issues in evaluation by success rates, recent works (Chen et al., 2021; Liu et al., 2024b,d; Hsiao et al., 2022) introduce benchmarks to assess VLMs' granular performance in web scenarios. Particularly, VisualWebBench (Liu et al., 2024d) focuses on tasks such as Web captioning/QA, Action Grounding/Prediction, etc. Moreover, WebQuest (Wang et al., 2024a) extends the above single web page evaluations to web page sequences. They propose Multi-Screen QA, which requires gathering information across multiple webpages, and Trace QA, asking about the entire sequence viewed by the user when browsing the web. However, their questions focus on extracting information across multiple web pages instead of reasoning or action prediction. In summary, the above works focus on two categories: 1). Perception, where VLMs process the textual and visual content of web pages to answer direct or complex information extraction questions; 2). Single-Step Action, where VLMs determine the nature and the location of action on the user screen. This lack of focus on Planning motivated our fine-grained investigation into such capabilities. We devise novel questions, experiment formulations and examine 19 VLMs to derive our analysis. To our knowledge, this is the first work to analyze VLMs' nuanced planning capabilities in web environment.

3 Experiment Setup

Overview: In our experiments (Sects. 4 and 5), we test VLMs on four groups of questions as discussed in Sec. 1 and Fig. 2 and derive model responses by prompting the model to solve multiple-choice questions (MCQs) with randomly shuffled options and one correct choice labeled. In Sec. 4.1, we also evaluate the model in a non-MCQ QA setup for comparison. Moreover, for each experiment, we prompt VLMs with direct instructions (Instruct) and chain-of-thought (CoT) reasoning (Kojima et al., 2023).
VLMs used: Our experiments evaluate 19 VLMs including general multi-modal and GUI-specific models. Considering the importance of practical-size web agents (Appx. K), for general VLMs, we choose Qwen2-VL (2B, 7B, 72B) (Wang et al., 2024b), Llava-1.6 (7B, 13B, 34B) (Liu et al., 2024a), Llava-One-Vision (0.5B, 7B) (Li et al., 2024a), Minicpm-V (Yao et al., 2024), Minicpm-o (MiniCPM-o Team, 2025), InternVL2-MPO (8B) (Wang et al., 2024c), DeepSeek-VL (Lu et al., 2024), GLM-4V (Lu et al., 2024), Idefics3 (Jiang et al., 2024), and Phi-3.5 Vision (et. al., 2024b), and for GUI-specific models, we consider Ferret-UI (2B, 8B) (You et al., 2024), UIX-Qwen2 (Liu et al., 2024c), and CogAgent (Hong et al., 2024). We utilize a maximum of 4 A4500 GPUs.

Model	Dataset1 - **MM-M2W**								Dataset2 - **GUIACT**								**Avg**
	MCQ				Non-MCQ				MCQ				Non-MCQ				
	Instruction		CoT		Instruction		CoT		Instruction		CoT		Instruction		CoT		
	Order1	Order2	Order1	Order2	Order1	Order2	Order1	Order2	Order1	Order2	Order1	Order2	Order1	Order2	Order1	Order2	
Qwen2-VL 2B	50.14	49.95	48.95	50.09	87.21	5.35	78.29	11.99	74.86	15.71	48.46	51.32	43.35	49.55	71.23	14.55	46.94
Qwen2-VL 7B	72.1	2.9	75.0	6.34	53.51	30.32	24.87	55.99	61.36	11.09	62.18	22.24	14.33	50.69	11.72	72.71	39.21
Qwen2-VL 72B	81.26	18.58	76.52	43.89	81.36	27.35	77.0	47.17	79.64	44.62	89.77	82.64	85.4	57.95	88.69	72.41	65.89
Minicpm	61.04	34.44	51.9	49.95	62.53	14.96	78.25	16.25	60.35	24.01	50.04	53.2	7.16	51.7	60.15	39.18	44.69
Phi 3.5	63.62	8.02	77.85	16.22	81.96	1.28	96.43	4.75	45.84	4.4	54.76	17.67	20.75	6.28	48.81	22.65	35.71
IDEFICS3	100.0	0.0	97.12	1.11	95.93	2.08	98.01	1.68	95.23	3.41	81.44	11.04	33.48	28.72	93.84	9.99	40.88
DeepSeek VL	100.0	0.0	68.83	27.63	100.0	0.0	98.41	8.9	100.0	0.0	83.79	14.16	97.97	0.25	98.31	1.3	31.41
Llava OV 0.5B	51.43	49.0	77.89	20.81	51.13	44.3	96.33	4.06	55.2	43.98	55.71	42.23	100.0	0.0	97.97	2.4	43.34
Llava OV 7B	72.34	17.59	79.28	23.38	25.66	48.66	85.62	12.38	83.84	7.78	80.93	20.01	0.03	77.81	55.79	16.78	44.24

Table 1: **Evaluation on Temporal Ordering Task.** We show the mean accuracy of various VLMs on two datasets across different configurations of the temporal ordering task. Pic1 and Pic2 indicate which the correct answer is. We observe most models fail to exceed the random guess baseline of 50%, with only Qwen2-VL 72B consistently surpassing it at 65.89%. The results highlight VLMs' reliance on spurious heuristics, i.e., the order of options, rather than genuine temporal understanding.

Response Evaluations: We parse the VLM output and measure accuracy against the ground truth of the questions as our metric. While we typically use deterministic regular expressions for parsing, some models did not follow the output format, for which we use Llama3-8B (et. al., 2024a) for parsing (see Appx. F). As additional baselines, we also conduct a human study with at least 15 raters on 15 randomly sampled questions (with the same input and instruction for VLMs) per test. More details about human study are in Appx. G.

4 Temporal Understanding Analysis

Understanding temporal relations is a fundamental aspect and prerequisite for effective planning. For web agents, comprehending the sequential order of events (e.g., "adding to shopping cart before checking out") and reasoning about how a webpage's state changes through a series of interactions is critical for making informed plans and decisions. In this section, our objective is to evaluate the temporal understanding of VLMs through two hypotheses: 1). Temporal Ordering: *Can VLMs identify which webpage screenshot/state comes earlier in the user's trajectory for a given task (e.g., filling out form comes before submitting it)* 2). Future State Prediction: *Can VLMs determine whether two sequences of actions lead to same future states?*

4.1 Temporal Ordering

Motivation: As agents often process sequences of prior state screenshots to predict subsequent actions, understanding the order of webpage states during navigation is crucial to reason about the dynamics in web tasks. This process requires agents to inherently understand task trajectories (e.g., during online shopping) and infer the sequential ordering of screenshots to make decisions. To test this requirement, we assess whether VLMs can understand the temporal relationship of screenshots representing different steps in a task.

Experiment Setup: In this test, we provide VLM with a task instruction, two webpage states (or screenshots), and the objective to determine which screenshot represents an earlier step in the trajectory to solve the task. We use Multimodal Mind2Web (MM-M2W) (Zheng et al., 2024) and the web-environment subset of GUIAct (Chen et al., 2024), both of which consist of a task and a ground-truth trajectory comprising T actions $\{a_t\}_{t=0}^{T}$ and corresponding webpage screenshots $\{i_t\}_{t=0}^{T}$. In these datasets, i_0 (or initial webpage state) is typically the homepage of a website and i_T (or final webpage state) is typically when task is completed. Therefore, for a clear and unambiguous test, we always use i_0, i_t for inputs, hypothesizing that presenting the initial and final webpage states is the easiest question to determine which comes earlier. For completeness, we include further discussion and ablation on randomly sampled i_j, i_k in Appx. C, which yields similar findings.

To prepare inputs for VLMs, we label the screenshot images with IDs "Picture 1" and "Picture 2", and the ordered input image tuple as (Picture 1, Picture 2). To mitigate the influence of prompt variations (specifically input orders), we create two broad configurations for each screenshot pair. In the first setting (referred to as Order1), the input image tuple is (start state i_0, end state i_T), and the expected answer is "Picture 1 comes earlier." (e.g., Fig. 2(a)); in the second configuration (referred to as Order2), the input image tuple is (end state i_T, start state i_0), and the expected answer is "Picture 2 comes earlier." For each of the above two configurations, we prompt the model in MCQ and non-MCQ format as described in Sec. 3. Each data point thus generates four question types, enabling a comprehensive evaluation of the model's ability to reason about temporal relationships.

Result - Poor sense of temporal relation: We observe in Tab. 1 that the average performance (the rightmost column) of almost every model fails to surpass the random guess performance of **50%** while a human rater achieves a performance of **about 96.2%**. Particularly, only Qwen2-VL 72B, with a mean accuracy of **65.89%** across 16 configurations, exhibits a consistent performance above random guess but still remains significantly lower than human performance. Additionally, we observe no significant performance variations between MCQ and non-MCQ versions in Tab. 1.

Finding - Spurious preference for the first input image: Moreover, we observe that irrespective of the order of images, VLMs tend to prefer the answer "Picture 1 comes earlier in the trajectory" after logically assessing both images before making a decision. Specifically, the mean performance difference across all models, between cases where "Picture 1" is the correct answer and cases where "Picture 2" is the correct answer, is 46.64% for MCQ and 41.66% for non-MCQ. We further study this in Appx. I through some qualitative examples, and find that VLMs tend to rationalize their answer of "Picture 1" with untenable reasons. These examples highlight VLMs' reliance on spurious heuristics over genuine temporal understanding, increasing their failures in temporal ordering tests.

4.2 Future State Prediction

Motivation: A critical aspect of temporal understanding and planning is imagining webpage state changes and predicting outcomes from action sequences. Particularly, a model must reason about the final states resulting from different action sequences to figure out the most effective course of action for achieving the goal. Although previous works briefly discuss next-state prediction (see Sec. 2), understanding outcomes over a series of actions remains underexplored for VLMs. To address this, we propose a simple formulation and examine whether VLMs can predict if different action plans lead to the same or distinct final states.

Experiment Setup: This test requires the model to evaluate two proposed action sequences, provided with an initial state screenshot i_0, and determine whether the future state i_t achieved by these plans are identical. To construct this evaluation, we utilize the data from MiniWob++ environment (Liu et al., 2018) to generate user interface screenshots, and two candidate action plans per screenshot, representing common web interactions involving UI elements such as checkboxes, textboxes, sliders (see Fig. 2(b) and more in Appx. A). To analyze the impact of trajectory length on planning capacity, we generate data samples with short-horizon (1-2 action steps) and long-horizon (3-4 action steps) plans. Although longer horizons (e.g., 6-8 steps) are possible, we limit our exploration to 1-4 steps, which already pose significant challenges for VLMs. For reliability, each data point is manually verified, including validating the logical coherence of action plans and the clarity of resulting states.

Result - Failure in predicting future states: We present our results in Tab. 2. While a random baseline would achieve approximately 50%, most models either perform below or near random guess in both short- and long-horizon scenarios. Notably, for the CoT setting in longer-horizon tasks, only (Qwen2 VL 72B) exceeds 75%. Meanwhile, human performance in these tasks is **97**%.

Finding - Easier to reason on shorter horizons: We see a trend where all VLMs, except CogAgent, perform better on short-horizon than long-horizon plans. While the accuracy gain varies, some models, such as Llava 1.6 - 34B, show improvements of 32% over their long-horizon results. This highlights the persistent limitations of current VLMs and raises concerns about their reliability due to inconsistencies across different scenarios.

5 Plan Assessment Analysis

In this section, we investigate more complex reasoning capabilities: *Can VLMs select the right plan to achieve a specific goal under real-world and challenging synthetic edge-case scenarios?* While the previous temporal analysis focuses on understanding the relationship between steps in a sequence, plan selection requires using this understanding to choose the optimal course of actions.

Model	Short Horizon		Long Horizon		Avg
	Instruct	CoT	Instruct	CoT	
Qwen2-VL 2B	55.68	53.35	50.42	49.36	52.2
Qwen2-VL 7B	66.69	76.33	67.45	48.38	64.71
Qwen2-VL 72B	98.29	97.35	76.88	67.23	84.94
Minicpm	51.13	62.55	50.0	35.57	49.81
Minicpm-o	57.17	68.43	57.82	48.83	58.06
Phi 3.5	67.52	66.96	45.11	47.14	56.68
IDEFICS3	34.67	79.57	50.07	50.49	53.7
DeepSeek VL	51.28	57.52	43.26	45.8	49.46
Llava OV 0.5B	50.78	44.28	49.28	44.01	47.09
Llava OV 7B	56.04	76.46	50.11	52.29	58.72
InternVL2-MPO	37.47	86.97	54.53	59.74	59.68
Glm	54.67	61.33	49.77	50.45	54.06
Llava 1.6 7B	65.27	64.18	40.61	48.63	54.67
Llava 1.6 13B	50.0	62.49	50.0	45.7	52.05
Llava 1.6 34B	80.03	76.39	46.17	53.78	64.09
Ferret-UI-Llama	46.76	49.99	66.41	49.85	53.25
Ferret-UI-Gemma	59.32	6.18	49.03	5.95	30.12
UIX-Qwen2	50.82	62.67	40.11	46.72	50.08
CogAgent	50.0	35.65	50.0	43.23	44.72

Table 2: **Evaluation on State Prediction Task in Sec. 4.2.** We show the accuracy of VLMs on the short- and long-horizon splits. While overall performance is suboptimal, models show improved accuracy with shorter horizons, underscoring their limitations in multi-step reasoning.

Model	Shuffle Perturb.		Semantic Perturb.		Avg
	Instruct	CoT	Instruct	CoT	
Qwen2-VL 2B	54.02	22.29	72.41	27.58	44.08
Qwen2-VL 7B	57.24	50.57	90.57	71.72	67.53
Qwen2-VL 72B	81.83	75.4	95.63	90.11	85.74
Minicpm	52.39	37.01	85.97	64.36	59.93
Minicpm-o	48.96	39.31	80.91	75.17	61.09
Phi 3.5	30.11	32.41	69.19	60.45	48.04
IDEFICS3	40.69	36.09	84.13	60.0	55.22
DeepSeek VL	24.13	23.67	26.89	28.73	25.85
Llava OV 0.5B	24.13	12.18	25.57	15.4	19.32
Llava OV 7B	55.17	47.12	89.88	69.65	65.46
InternVL2-MPO	45.05	22.75	90.11	68.73	56.66
Glm	30.34	35.86	70.8	67.81	51.2
Llava 1.6 7B	26.95	30.02	38.7	47.11	35.7
Llava 1.6 13B	28.34	24.88	59.44	45.85	39.62
Llava 1.6 34B	39.86	43.41	85.94	73.27	60.61
Ferret-UI-Llama	23.67	20.68	22.98	21.65	22.24
Ferret-UI-Gemma	11.95	22.06	9.88	20.04	15.98
UIX-Qwen2	57.7	29.65	67.35	43.21	49.48
CogAgent	23.41	23.9	26.25	26.72	25.07

Table 3: **Evaluation on Plan Selection in Real Web Environment. in Sec. 5.1** We show the accuracy of VLMs with two types of MCQ choices (Sec. 5.1). Many VLMs perform under 50%. Models perform better to distinguish semantically different options than options with shuffled orders.

Model	Instruct	CoT	Avg
Qwen2-VL 2B	37.83	35.47	36.65
Qwen2-VL 7B	49.56	47.84	48.7
Qwen2-VL 72B	88.75	76.25	82.5
Minicpm	33.77	33.82	33.8
Minicpm-o	47.76	37.35	42.56
Phi 3.5	32.85	26.18	29.52
IDEFICS3	46.2	55.87	51.04
DeepSeek VL	18.33	28.17	23.25
Llava OV 0.5B	19.45	21.65	20.55
Llava OV 7B	37.55	38.41	37.98
InternVL2-MPO	57.69	49.15	53.42
Glm	46.77	42.23	44.5
Llava 1.6 7B	29.33	40.25	34.79
Llava 1.6 13B	26.09	31.96	29.02
Llava 1.6 34B	40.35	29.72	35.04
Ferret-UI-Llama	20.83	22.61	21.72
Ferret-UI-Gemma	11.72	13.28	12.5
UIX-Qwen2	22.65	14.46	18.56
CogAgent	26.34	21.08	23.71

Table 4: **Evaluation on Plan Selection in Edge Cases in Sec. 5.2** We observe a similar trend to Tab. 3 here. Moreover, CoT consistently hurts the model performance compared to the direct instruction.

This transition represents a critical step in evaluating whether VLMs can function as reliable web agents. Specifically, each test example consists of a goal description, a screenshot of current webpage state, and possible action plan candidates presented in an MCQ format. The model must identify the single correct plan to achieve the desired goal.

To analyze whether VLMs can reliably reason and select plans across diverse contexts, we conduct two tests. First, we evaluate real-world web use cases using tasks and scenarios from actual website data. Second, we leverage synthetic edge cases designed to challenge VLMs in controlled environments, offering a more comprehensive assessment of their reasoning abilities.

5.1 Real Web Environment Study

Experiment Setup: To test VLMs in real-world web environments, we evaluate whether VLMs can choose the correct plan from four MCQ options, given the current webpage screenshot and task instruction. Specifically, we curate the real-web dataset from the MM-M2W dataset (Zheng et al., 2024), where each data point consists of a task instruction and a human-annotated trajectory to solve that task. To leverage this data, we first perform a few modifications. We begin by removing samples where the action sequence is longer than 10 steps to align with VLMs' reasoning limitations (Nagar et al., 2024; Małkiński et al., 2024). Then, to make the dataset's provided action descriptions (e.g., "Target element: [button] Search, Action: CLICK") more natural to VLMs (e.g., "Click on the search button"), we paraphrase them with GPT-4o (OpenAI, 2024). To ensure data quality, we manually inspect and refine each sample.

We construct the final test MCQ questions as follows: For each task instruction g, we sample the web page screenshot (i.e., state) i_t from its solution trajectory, where i_t is 4-5 action steps away from reaching the task's end state (or goal completion). These action steps will now serve as the ground-truth correct answer for MCQ. By limiting the reasoning to the next 4–5 steps, we simplify the task for a reasonable chance of success. Incorrect MCQ options are then populated using the following two strategies:

○ **Shuffling Perturbation:** Given that a plan is a sequence of actions where the next action depends on the previous actions, this configuration creates incorrect options by violating the sequential flow of actions. Particularly, we create incorrect options by entirely or partially reversing the correct action sequence, e.g., for the task "Buy a Shampoo on Amazon" and the plan ["Visit Amazon", "Search Shampoo", "Add to Cart", "Order"], an incorrect option would be ["Order", "Add to Cart", "Visit Amazon", "Search Shampoo"] (see Fig. 26c).

○ **Semantic Perturbation:** Given an action plan for a task, this configuration generates incorrect options by inserting irrelevant actions from a different task into the current task's action plan. Particularly, for the task "Buy a Shampoo on Amazon" and correct action plan ["Visit Amazon", "Search Shampoo", "Add to Cart", "Order"], an incorrect option would be ["Visit Amazon", "Search flights to New York", "Add to Cart", "Order"] (see Fig. 26d). With manual validation, we obtain 435 high-quality samples per strategy, as detailed in Appx. B.

Result - Lack of genuine understanding of plans: The experiment results are presented in Tab. 3. Under the Shuffling Perturbation setting, while most VLMs outperform random guess performance (i.e., 25%), their accuracy remains significantly lower than that of human raters (i.e., 90%), with only 3 models exceeding 50%. In contrast, in the Semantic Perturbation setting, many models achieve higher accuracy over 50%, and every model surpasses its own performance under Shuffling Perturbation (due to intuitiveness, we skip human study for the generally easier Semantic Perturbation setting). This suggests that while VLMs can differentiate plans by assessing whether each of their action is contextually appropriate for the given task (shown in Appx. I), they struggle to compare plans based on the difference between the temporal ordering of actions, which is critical for planning. Consequently, we anticipate a need to explicitly model action sequences and state changes in VLMs.

5.2 Edge Case Test Study

Motivation: Real-world data provides valuable insights but often lacks the diversity of scenarios that agents may encounter during web browsing. For instance, datasets like MM-M2W include human-annotated trajectories that typically start from website homepages, have human labels for each action, as well as represent "perfect" trajectories with minimal navigation errors and where each webpage's state is likely an 'ideal state' for possible actions (e.g., to fill a form, we define an 'ideal state' is an empty form). However, in practice, edge cases such as random initializations (e.g., forms have pre-filled fields) or error states (e.g., checking the wrong checkbox) are common. While humans can recover from these, we explore whether VLMs can do the same. Thus, motivating our development of synthetic edge case data with controlled variations, focusing on recoverable errors.

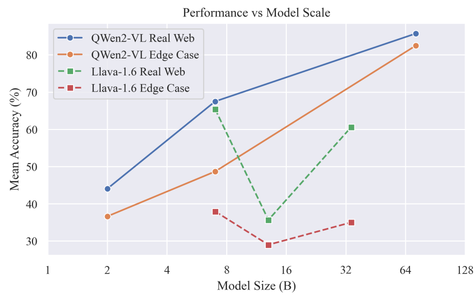


Figure 3: **Performance v/s Model Scales.** We plot average performance of Qwen2-VL and Llava-1.6 family over all settings in real web environment (Sec. 5.1) and edge case experiments (Sec. 5.2). We find that scaling does not automatically help every model.

Experiment Setup: We again leverage MiniWob++, focusing on UI elements including checkboxes, textboxes, sliders, radio-buttons, and buttons. For a given task (e.g., "Check the 1st and 2nd checkbox"), we algorithmically perform actions that modify the empty form state to introduce random initializations or errors (e.g., toggling the 3rd checkbox), and then generate four action plans, only one of which correctly solves the task (see Appx. A). To ensure high-quality data, we manually validate the correctness of the questions, and the existence of a single correct option.

Result - Worse performance on edge cases: From Tab. 4, we observe that most models perform significantly poor, with accuracy below 50% for both Instruct and CoT variants. Meanwhile, human raters achieve 99.2% accuracy for these tasks. The performance is even poorer for GUI-specific models (Ferret-UI, UIX-Qwen2, CogAgent), with a maximum accuracy of only 26%. Moreover, we observe that the highest-performing Qwen2VL 72B model suffers a 12% performance drop in its CoT variant. This reinforces concerns if models rely on spurious correlations rather than genuine understanding to arrive at correct answers.

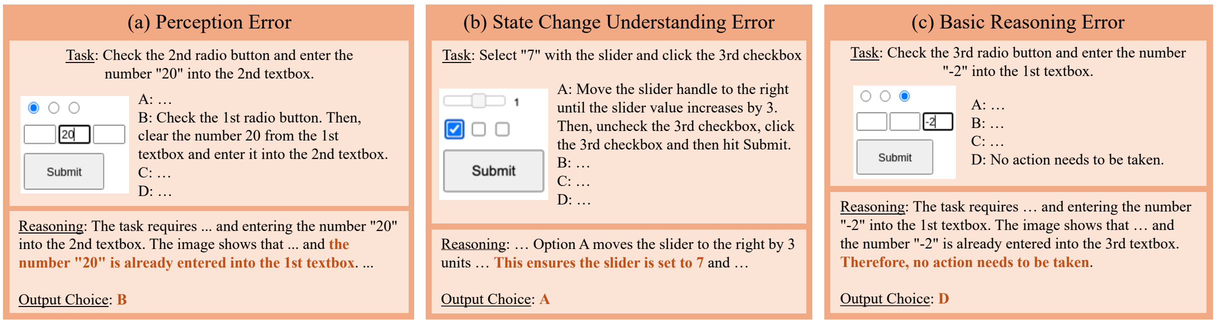


Figure 4: **Error case study.** We demonstrate three common error paradigms observed in our experiments.

6 Discussion and Analysis

In this section, we highlight the consistent trends observed across our experimental settings.

Do these skills lead to task success? We compare the task success of three models—Qwen2-VL-7B, LLaVA-1.6, and Phi-3.5— from the Mind2Web-Live benchmark (Pahuja et al., 2025), and their performance across our planning skills. As detailed in Appx. D, model's capabilities in our planning skills are correlated with the task success rate.

Is CoT reasoning always helpful? In Tabs. 1 to 4, we explore the impact of CoT reasoning on the performance in each experiment. Interestingly, we did not observe sufficient gains resulting from CoT. Specifically, while some models occasionally showed reduced performance due to CoT, others consistently struggled to adopt it effectively.

Are bigger models better than their smaller versions? From Fig. 3, we find that while Qwen2-VL family improves in performance, Llava 1.6-34B is not significantly better than its smaller counterparts, contrary to the findings of (Liu et al., 2024d) - scaling improves perception-related web tasks. Thus, it is crucial to investigate how factors beyond scaling—such as architecture and pre-training—impact VLMs' reasoning and planning abilities on the web.

Do GUI-specific models outperform general VLMs? Although previous studies show that GUI-specific models outperform generalist VLMs in GUI navigation and comprehension tasks (Rawles et al., 2024; Cheng et al., 2024), they significantly underperform on our tests in Tabs. 2 to 4 (last three rows). Similar to (Liu et al., 2024d), we found this behavior is likely due to over-fitting to the GUI settings, which affects their general reasoning and instruction-following capability. We support this using qualitative examples of GUI-Models in Appx. I, demonstrating their poor output quality.

Case Study: We present examples in Fig. 4 to better understand the errors made by the model. This section focuses on the results of Qwen2-VL 72B in Sec. 5.2, with additional results provided in Appx. I. Our analysis reveals three most common errors: 1). Perception Errors (Fig. 4a), where incorrect perception of details becomes bottleneck in reasoning, 2). State Change Understanding (Fig. 4b), where the model fails to imagine future states, 3). Basic Reasoning (Fig. 4c), where the model fails to make straightforward logical inferences.

7 Conclusion

In this work, we introduce the first evaluation frameworks aimed at understanding the fine-grained skills required for VLMs to plan in web environments. While prior research has largely focused on webpage perception and task success metrics, such evaluations overlooks the necessity in thoroughly assessing the planning capability required for reliable web agents to plan. To address this gap, we carefully design critical questions encompassing planning skills such as temporal relations over state trajectory, temporal prediction of future states, and plan assessment for identifying effective plans. By carefully re-purposing existing data, generating synthetic data and validating by humans, we create four high-quality tasks and benchmark 19 state-of-the-art VLMs.

Our findings reveal that current VLMs are far from achieving human-like planning capabilities, and hardly above random guessing in many evaluations, indicating fundamental deficiencies in reasoning about web dynamics. Specifically, we observe that models struggle with temporal reasoning and plan assessment, highlighting their limitations in synthesizing sequential web information. Additionally, common strategies, such as CoT reasoning, increasing model scale, and training on GUI-specific datasets, do not consistently lead to improvements in fine-grained planning tests. These findings underscore the pressing need for more sophisticated training and evaluation paradigms.

Limitations

Despite our contributions, we acknowledge several limitations. Due to resource constraints, we are yet to evaluate proprietary models (e.g GPT, Claude, Gemini), which may exhibit different planning capabilities. Additionally, our reliance on repurposed and synthetic data, while effective in establishing structured evaluations, limits the coverage of diverse real-world scenarios. Given the complexity and variability of web interactions, we only address a key subset of important planning challenges. Nevertheless, we hope that our study serves as a foundation for future research, encouraging further exploration of VLMs' planning capabilities and reliablitiy as web agents.

References

Wentong Chen, Junbo Cui, Jinyi Hu, Yujia Qin, Junjie Fang, Yue Zhao, Chongyi Wang, Jun Liu, Guirong Chen, Yupeng Huo, Yuan Yao, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. Guicourse: From general vision language models to versatile gui agents.

Xingyu Chen, Zihan Zhao, Lu Chen, JiaBao Ji, Danyang Zhang, Ao Luo, Yuxuan Xiong, and Kai Yu. 2021. WebSRC: A dataset for web-based structural reading comprehension. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4173–4185, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. 2024. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9313–9332, Bangkok, Thailand. Association for Computational Linguistics.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Aaron Grattafiori et. al. 2024a. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Marah Abdin et. al. 2024b. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Akash Ghosh, Arkadeep Acharya, Sriparna Saha, Vinija Jain, and Aman Chadha. 2024. Exploring the frontier of vision-language models: A survey of current methodologies and future directions. *Preprint*, arXiv:2404.07214.

Yu Gu, Boyuan Zheng, Boyu Gou, Kai Zhang, Cheng Chang, Sanjari Srivastava, Yanan Xie, Peng Qi, Huan Sun, and Yu Su. 2024. Is your llm secretly a world model of the internet? model-based planning for web agents. *Preprint*, arXiv:2411.06559.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024. WebVoyager: Building an end-to-end web agent with large multimodal models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6864–6890, Bangkok, Thailand. Association for Computational Linguistics.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxuan Zhang, Juanzi Li, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. 2024. Cogagent: A visual language model for gui agents. *Preprint*, arXiv:2312.08914.

Yu-Chung Hsiao, Fedir Zubach, Gilles Baechler, Victor Carbune, Jason Lin, Maria Wang, Srinivas Sunkara, Yun Zhu, and Jindong Chen. 2022. Screenqa: Large-scale question-answer pairs over mobile app screenshots. *arXiv preprint arXiv:2209.08199*.

Yuanhan Jiang, Yiyang Gu, Hervé Jégou, and Adam Lerer. 2024. Building and better understanding vision-language models: insights and future directions. *Preprint*, arXiv:2408.12637.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. 2024a. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *ACL*.

Jing Yu Koh, Stephen McAleer, Daniel Fried, and Ruslan Salakhutdinov. 2024b. Tree search for language model agents. *Preprint*, arXiv:2407.01476.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Lin Li, Guikun Chen, Hanrong Shi, Jun Xiao, and Long Chen. 2024b. A survey on multimodal benchmarks: In the era of large ai models. *Preprint*, arXiv:2409.18142.

Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. 2018. Reinforcement learning on web interfaces using workflow-guided exploration. In *International Conference on Learning Representations (ICLR)*.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024a. Llava-next: Improved reasoning, ocr, and world knowledge.

Junpeng Liu, Tianyue Ou, Yifan Song, Yuxiao Qu, Wai Lam, Chenyan Xiong, Wenhu Chen, Graham Neubig, and Xiang Yue. 2024b. Harnessing web-page uis for text-rich visual understanding. *Preprint*, arXiv:2410.13824.

Junpeng Liu, Tianyue Ou, Yifan Song, Yuxiao Qu, Wai Lam, Chenyan Xiong, Wenhu Chen, Graham Neubig, and Xiang Yue. 2024c. Harnessing web-page uis for text-rich visual understanding. *Preprint*, arXiv:2410.13824.

Junpeng Liu, Yifan Song, Bill Yuchen Lin, Wai Lam, Graham Neubig, Yuanzhi Li, and Xiang Yue. 2024d. Visualwebbench: How far have multimodal llms evolved in web page understanding and grounding? *Preprint*, arXiv:2404.05955.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. 2024. Deepseek-vl: Towards real-world vision-language understanding. *Preprint*, arXiv:2403.05525.

Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. 2024. A survey on vision-language-action models for embodied ai. *Preprint*, arXiv:2405.14093.

Mikołaj Małkiński, Szymon Pawlonka, and Jacek Mańdziuk. 2024. Reasoning limitations of multimodal large language models. a case study of bongard problems. *Preprint*, arXiv:2411.01173.

MiniCPM-o Team. 2025. MiniCPM-o 2.6: A GPT-4o-Level MLLM for Vision, Speech, and Multimodal Live Streaming on Your Phone. Accessed: January 31, 2025.

Aishik Nagar, Shantanu Jaiswal, and Cheston Tan. 2024. Zero-shot visual reasoning by vision-language models: Benchmarking and analysis. *Preprint*, arXiv:2409.00106.

OpenAI. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Vardaan Pahuja, Yadong Lu, Corby Rosset, Boyu Gou, Arindam Mitra, Spencer Whitehead, Yu Su, and Ahmed Awadallah. 2025. Explorer: Scaling exploration-driven web trajectory synthesis for multimodal web agents. *Preprint*, arXiv:2502.11357.

Jiayi Pan, Yichi Zhang, Nicholas Tomlin, Yifei Zhou, Sergey Levine, and Alane Suhr. 2024. Autonomous evaluation and refinement of digital agents. *Preprint*, arXiv:2404.06474.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. 2024. Androidinthewild: A large-scale dataset for android device control. *Advances in Neural Information Processing Systems*, 36.

Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. 2017. World of bits: An open-domain platform for web-based agents. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3135–3144. PMLR.

Segev Shlomov, Aviad Sela, Ido Levy, Liane Galanti, Roy Abitbol, et al. 2024. From grounding to planning: Benchmarking bottlenecks in web agents. *arXiv preprint arXiv:2409.01927*.

Maria Wang, Srinivas Sunkara, Gilles Baechler, Jason Lin, Yun Zhu, Fedir Zubach, Lei Shu, and Jindong Chen. 2024a. Webquest: A benchmark for multimodal qa on web page sequences. *Preprint*, arXiv:2409.13711.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024b. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Weiyun Wang, Zhe Chen, Wenhai Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Jinguo Zhu, Xizhou Zhu, Lewei Lu, Yu Qiao, and Jifeng Dai. 2024c. Enhancing the reasoning ability of multimodal large language models via mixed preference optimization.

Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. 2025. Aguvis: Unified pure vision agents for autonomous gui interaction. *Preprint*, arXiv:2412.04454.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. preprint. Webshop: Towards scalable real-world web interaction with grounded language agents. In *ArXiv*.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Keen You, Haotian Zhang, Eldon Schoop, Floris Weers, Amanda Swearngin, Jeffrey Nichols, Yinfei Yang, and Zhe Gan. 2024. Ferret-ui: Grounded mobile ui understanding withnbsp;multimodal llms. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXIV*, page 240–255, Berlin, Heidelberg. Springer-Verlag.

Ziniu Zhang, Shulin Tian, Liangyu Chen, and Ziwei Liu. 2024. Mmina: Benchmarking multihop multimodal internet agents. *Preprint*, arXiv:2404.09992.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. Gpt-4v(ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*.

A Creating Synthetic Data with MiniWob++

In this section, we describe more details and examples for the data we create for Sec. 4.2, Sec. 5.2 using the MiniWob++ environment.

Edge Case Test: As shown in Fig. 6, given a task (e.g "Check abs checkbox"), we first obtain a clean or fresh form which has not been interacted with. Next, we algorithmically perform a series of actions, where we randomly interact with a subset of form elements (e.g clicking one/two checkboxes). to generate our edge-case screenshots. This process results in an edge case screenshot where some incorrect elements may be clicked, or certain fields may have different pre-filled values, and VLMs have to figure out their way to navigate (or recover) through this altered state and solve the task. We perform the above across three MiniWob++ environments, each featuring a form with a distinct combination of elements - (1) checkboxes only, (2) a slider and a checkbox, (3) a textbox and a radiobutton (See Fig. 5). Finally, these environments correspond to three specific tasks for our Edge Case test, namely Task 1, Task 2, and Task 3, respectively (See Fig. 7).

Future State Consistency: Now, we leverage the form screenshots obtained above and algorithmically create three tasks (Task 1, Task 2, Task 3) to examine if VLMs can determine whether two sequences of action lead to the same future states (see Fig. 8 for long-horizon action plan scenarios and see Fig. 9 for shorter ones).

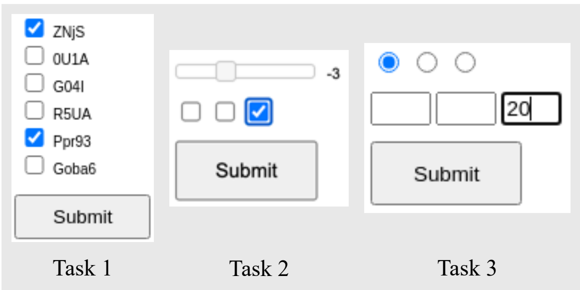


Figure 5: **Example Synthetic Data from MiniWob++.** We leverage the MiniWob++ environment to generate three groups of tasks: 1). a list of checkboxes, 2). a slide with three checkboxes, and 3). three radio-buttons with three text boxes. All tasks contain a "Submit" button.

B Constructed Options for Real Web Plan Assessment

We use the snippet in Fig. 10 to create options for our shuffling perturbation, and use the snippet in Fig. 11 to create the semantic perturbation.

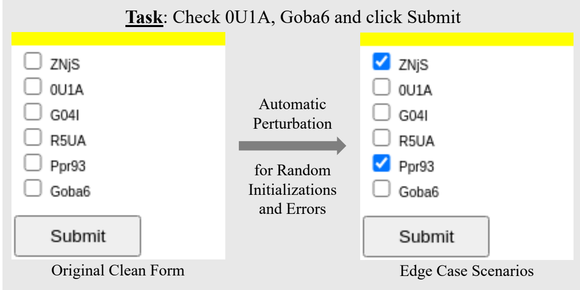
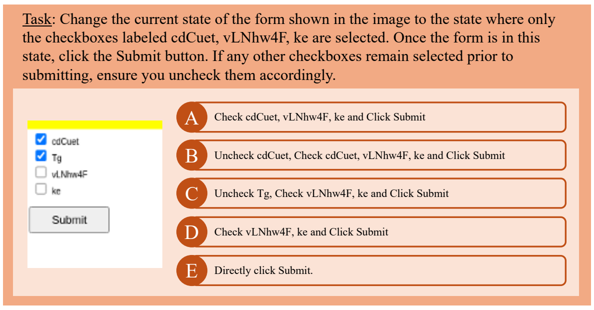
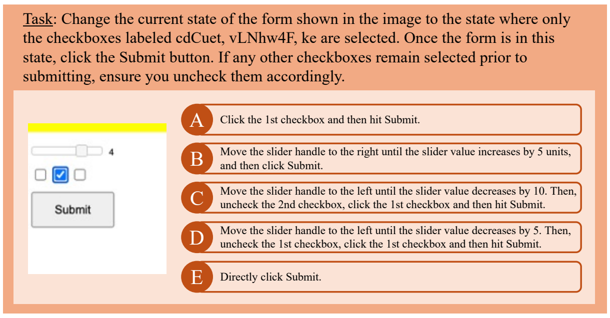


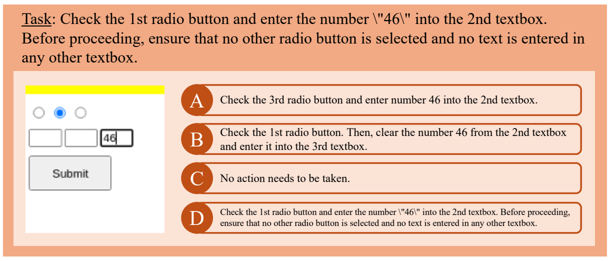
Figure 6: Illustrating one of our algorithmic interactions to perturb the original clean form state.



(a) Task 1



(b) Task 2

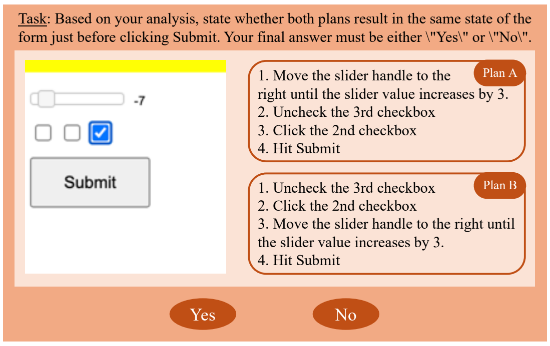


(c) Task 3

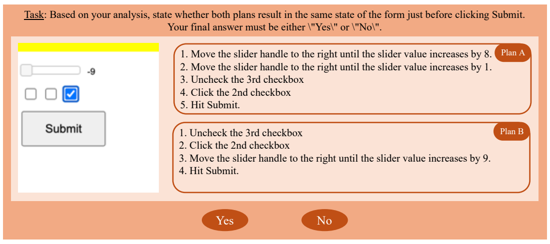
Figure 7: Task examples for Edge Case Test Study

C Choice of States in the Temporal Ordering Setup

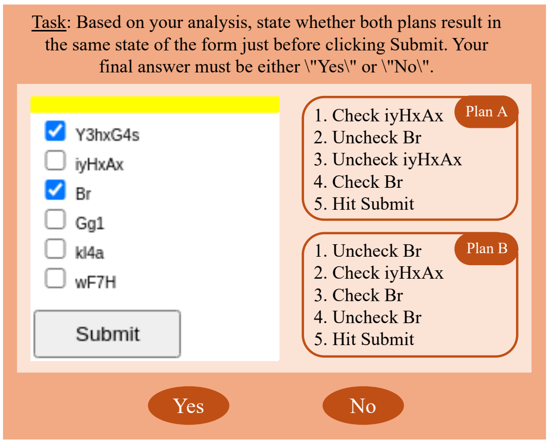
As discussed earlier, our design in the temporal ordering experiment focuses on transitions between the initial state i_0 and the final state i_t to reduce ambiguity. We now present a discussion for our choice along with the ablation experiments after selecting random states i_j, i_k.



(a) Task 1

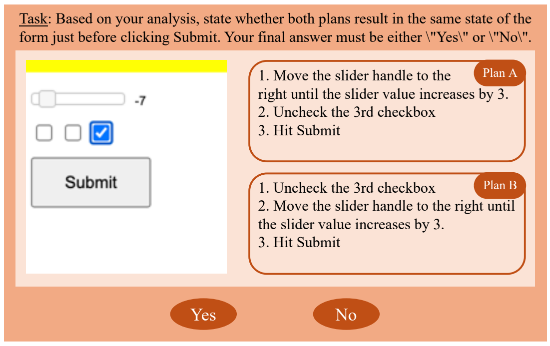


(b) Task 2

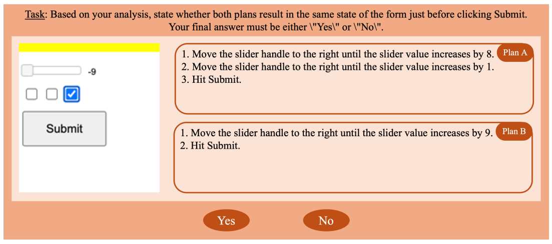


(c) Task 3

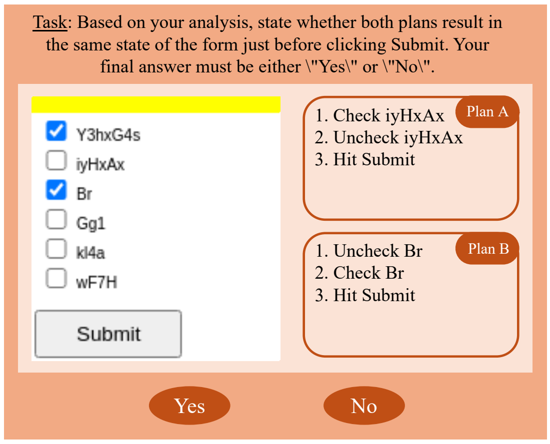
Figure 8: Task examples for Future State Prediction study – Long Horizon i.e plans have (3-4 action steps before submitting the form)



(a) Task 1



(b) Task 2



(c) Task 3

Figure 9: Task examples for Future State Prediction study – Short Horizon i.e plans have (1-2 action steps before submitting the form)

```
def shuffling_perturbation(arr):

    n = len(arr)

    # Custom shuffling logic for arrays of length 4
    if n == 4:
        shufflings = [
            arr[:],  # Original order: [pos1, pos2, pos3, pos4]
            arr[::-1],  # Reversed order: [pos4, pos3, pos2, pos1]
            [arr[3], arr[2], arr[0], arr[1]],  # Custom order: [pos4, pos3, pos1, pos2]
            [arr[2], arr[3], arr[1], arr[0]]  # Custom order: [pos3, pos4, pos2, pos1]
        ]
    # Custom shuffling logic for arrays of length 5
    elif n == 5:
        shufflings = [
            arr[:],  # Original order: [pos1, pos2, pos3, pos4, pos5]
            arr[::-1],  # Reversed order: [pos5, pos4, pos3, pos2, pos1]
            # Custom order: [pos5, pos4, pos3, pos1, pos2]
            [arr[4], arr[3], arr[2], arr[0], arr[1]],
            # Custom order: [pos4, pos3, pos2, pos1, pos5]
            [arr[3], arr[2], arr[1], arr[0], arr[4]]
        ]
    else:
        # Return None if array length is not 4 or 5
        return None

    # Randomly shuffle the generated shufflings
    random.shuffle(shufflings)

    return shufflings
```

Figure 10

```

# Get the plan to be perturbed based on the current index
plan_to_be_perturbed = all_plans_list[idx]

# sampled_id_for_random_plan --> this is an index from the list of plans other than idx
ref_plan = random.sample(all_plans_list[sampled_id_for_random_plan], 1)[0]

# Replace one action of the plan_to_be_perturbed with the sampled ref_plan
plan_to_be_perturbed[random.randint(0, len(plan_to_be_perturbed) - 1)] = ref_plan

# Append the perturbed plan to the list of semantic-perturbed plans
plan_choice_semantic_perturbed.append(plan_to_be_perturbed)
```

Figure 11

Discussion. Our original choice addresses challenges of noise associated with the random sampling of intermediate webpage states. For example, in tasks like *"Buy Louis Vuitton on Amazon,"* users scroll through search results dynamically, producing intermediate screenshots that reflect varying, non-sequential segments of the page. These variations, which do not follow a consistent semantic order, complicate efforts to determine which state occurs earlier. Similarly, in tasks such as *"Add toilet paper and laundry detergent to Amazon cart,"* the order of intermediate screenshots can plausibly vary depending on user behavior, making definitive temporal labels infeasible. Thus, creating a reliable evaluation data in the random way necessitates the need for manually inspecting and filtering the ambiguous intermediate states and the entire generated data, which is out of scope for our work.

Ablation Results. For the sake of completeness, we now include evaluation results with random states i_j, i_k. From Table 5, we can observe that for both the datasets MM-M2W, GUIAct, the performance of chosen VLMs (MiniCPM, Qwen2-VL-7B) is poor and not very substantiallyy better than the random guess performance of 50%. Although the potential for the aforementioned issues makes it harder to draw trends from these numbers, we can still observe the VLMs to struggle similarly for the intermediate states.

Model	MM-M2W		GUIAct	
	Instruct	CoT	Instruct	CoT
Minicpm	46.28	48.26	51.65%	59.35%
Qwen2-VL 7b	44.94	44.72	54.28%	54.64%

Table 5: Comparison of performance on MM-M2W and GUIAct datasets using random intermediate states.

D Correlation Between Planning Skill Performance and Task Success Rate

To investigate the relationship between planning skill performance and overall task success, we analyze the results of three models—Qwen2-VL-7B, LLaVA-1.6, and Phi-3.5—on the Mind2Web-Live benchmark (Pahuja et al., 2025). We evaluate these models across three core planning skills: *Future State Prediction*, *Plan Selection (Real World)*, and *Edge Case Detection*, along with their task success metrics, including average step success rate, completion rate, and full task success rate. The results are summarized in Table 6.

Model	Future State Prediction	Plan Selection (Real World)	Edge Case Detection	Planning Avg	Avg Step SR	Completion Rate	Full Task SR
LLaVA-1.6	35.70	54.67	34.79	41.72	32.0	30.3	4.8
Phi-3.5	56.68	48.04	29.52	44.75	44.0	34.0	13.3
Qwen2-VL-7B	64.71	67.53	48.70	60.31	45.3	40.2	19.3

Table 6: Comparison of planning skill evaluation and task success metrics across models on the Mind2Web-Live benchmark.

As shown in Table 6, the average planning skill scores for LLaVA-1.6, Phi-3.5, and Qwen2-VL-7B are 41.72, 44.75, and 60.31, respectively. These scores correlate with each model's task success rates: LLaVA-1.6 achieves a full task success rate of 4.8%, while Phi-3.5 and Qwen2-VL-7B achieve 13.3% and 19.3%, respectively. The ranking of models by average planning ability closely mirrors their ranking in task-level performance metrics, suggesting a strong relationship between planning skill and downstream task effectiveness. This finding reinforces the value of skill-level evaluations in understanding and predicting model performance in real-world web environments.

E Comparison with Proprietary Models

Given resource constraints, we were unable to run the complete set of experiments on large-scale proprietary models. To demonstrate the value of our experimental setup, we conducted smaller-scale evaluations using **GPT-4o-mini** across two representative tasks: *Temporal Ordering* and *Plan Selection in Real Web Environments*.

For the Temporal Ordering task, we select the GUIAct data and sample a subset of 500 questions from the full evaluation set, whereas for Plan Assessment, we evaluate on the Shuffle Perturbation setting. The results are shown in Table 7.

Phase	Instruct	CoT
Order1	40.6	41.6
Order2	57.8	59.6

Table 7: Temporal Ordering task performance (%) on GPT-4o-mini.

Prompt Type	Accuracy (%)
Instruct	49.66
CoT	48.97

Table 8: Plan Selection task performance (%) on GPT-4o-mini under the Shuffle Perturbation setting.

Our results with GPT-4o-mini reveal two key findings: (1) temporal prediction accuracy is only marginally better than random guessing (50%), and (2) performance on the plan selection task is weaker than that of many open-source models (see Table 3). These observations suggest that proprietary models do not necessarily outperform smaller open models on fine-grained web planning evaluations, highlighting the value of our task-specific benchmarking approach.

F Prompts

In this section, we include the following prompts:
Temporal Ordering: For simplicity, we only include the prompts for Non-MCQ settings. Note that we use the same prompt for the MCQ setting, differing only in the output format. Fig. 12 shows the Instruction prompt and Fig. 13 shows the CoT prompt.
Future State Prediction: Fig. 14 shows the Instruction prompt and Fig. 15 shows the CoT prompt.
Plan Assessment : Real Web Environment Fig. 16 shows the Instruction prompt and Fig. 17 shows the CoT prompt. We add action history to the prompt in Fig. 18 to evaluate the impact of the presence of history in Appx. H.
Plan Assessment : Edge Case Scenario Fig. 19 shows the Instruction prompt and Fig. 20 shows the CoT prompt.
Llama as a Judge Prompt Fig. 22, Fig. 21, Fig. 23 includes all the LLama prompts we use throughout our paper.

G Human Study Details

In this section, we describe the process we followed to obtain human performance for the tests mentioned in this paper.
Number of samples. We randomly sample 15 questions for Temporal Ordering, Plan Selection (Real Web Scenarios), and Plan Selection (Edge Cases). For Future State Prediction, which has Long and Short Horizon configurations, we sample 20 total questions to balance volunteer effort and question count. As noted in the main paper, of the two configurations for Plan Selection (Real Web)—Shuffling Perturbation and Semantic Perturbation—we conduct human study only for the harder Shuffling Perturbation due to constraints.
Instructions given to raters. The tests are presented to humans in the same format as the VLMs in our study, i.e., MCQs with 4/5 options. To ensure a fair comparison, we present the same instructions to both VLMs and human raters. Accordingly, we use the prompts outlined in Fig. 19, Fig. 14, Fig. 16, and Fig. 12.
Gathering Responses. Our raters included individuals from diverse backgrounds, including students across various degrees (Bachelor's, Master's, PhD) and fields (CS, ECE, Mathematics, Robotics, Nursing, Pharmacy, etc.), as well as Postdoctoral Researchers, Research Associates, and Technical Industry Professionals. Responses were collected via Google Forms. Depending on their availability, each rater participated in 1–4 studies, with 2 being the most common. This ensured a minimum of 15 responses per study.
Results. The performances of human raters are **96.2%** for Temporal Ordering, **97%** for Future State Prediction, **90%** for Plan Selection – Real Web Scenarios, and **99.2%** for Plan Selection – Edge Case Test. Qualitative feedback from raters revealed that most errors occurred due to rater fatigue of questions, rushing, or missing thorough reading of the task instruction and options.

H Impact of Action History in Future Plan Assessment

As described in Sec. 5.1, we use the MM-M2W dataset (Zheng et al., 2024), where each data point includes a task instruction and a human-annotated trajectory (a sequence of states/screenshots and actions at each state) to complete the task. We first select samples with action trajectories of length ≤ 10. To create a test question for each task instruction g, we sample a web page screenshot (or state) i_t from its solution trajectory, where i_t is 4-5 action steps (a_p) away from the task's end state (or goal completion). Thus, it can be understood that for sequences with lengths ≥ 5, there will be i_t steps leading from the start state (i_0, typically the homepage) to i_t. We refer to these as "previous actions." or "action history". In this ablation experiment, we investigate whether VLM performance in plan assessment improves when provided with "previous actions" as input, specifically testing if VLMs can better assess plans when given the history of actions. To construct prompts for this study, we adapt those used in Sec. 5.1, e.g, Fig. 18 is derived from Fig. 16.

Our results in Tab. 9 reveal a trend similar to the main paper table without action history. Notably,

Temporal Ordering (No-MCQ) (Instruction Prompting)

You are given a task along with two image screenshots representing steps in a trajectory that either solves the given question or helps find the relevant information required to answer it. Your goal is to determine which screenshot likely represents an earlier step in the trajectory.

An **earlier step in the trajectory** refers to a step that must logically or procedurally occur before the other. Consider dependencies, prerequisites, or actions that logically precede others. For example, if one step involves entering information and another involves confirming it, entering the information must logically come earlier. Use this principle to analyze the order of the screenshots.

Use the task to understand the context, and carefully examine each screenshot for any clues that suggest their order in the sequence. Consider all visible text, UI elements, and page structure to infer which screenshot likely comes earlier in the process.

Important Notes:
1. **Picture Labelling**: Refer to the input image screenshots as [Picture 1, Picture 2]. These labels are purely for identification and do not indicate any logical order or sequence in the trajectory. Your objective is to determine which picture represents the earlier step in the trajectory based on the given question description and careful analysis of the pictures.

2. **Relevance to the Task**: Both screenshots are directly related to either solving the question or finding relevant information to answer it. Do not assume that either image is irrelevant.

3. **Expert Trajectories**: These screenshots are taken from expert human trajectories that were executed to successfully solve the question. Trust that the screenshots reflect valid steps.

4. **Arbitrary Order**: The screenshots are presented in no particular order. Their arrangement does not indicate which step comes earlier in the trajectory.

Task:
{task}

Question:
Based on the two given images and the question, determine which image (or screenshot) represents an earlier step in the trajectory.

Output format:
Clearly state which picture represents the **earlier step** in the trajectory, "Picture 1" or "Picture 2".

Only output your answer choice ("Picture 1" or "Picture 2"). Do not include any reasoning, explanation, or additional text.

Figure 12

Temporal Ordering (No-MCQ) (Chain-of-Thought Prompting)

You are given a task along with two image screenshots representing steps in a trajectory that either solves the given question or helps find the relevant information required to answer it. Your goal is to determine which screenshot likely represents an earlier step in the trajectory.

An **earlier step in the trajectory** refers to a step that must logically or procedurally occur before the other. Consider dependencies, prerequisites, or actions that logically precede others. For example, if one step involves entering information and another involves confirming it, entering the information must logically come earlier. Use this principle to analyze the order of the screenshots.

Use the task to understand the context, and carefully examine each screenshot for any clues that suggest their order in the sequence. Consider all visible text, UI elements, and page structure to infer which screenshot likely comes earlier in the process.

Important Notes:
1. **Picture Labelling**: Refer to the input image screenshots as [Picture 1, Picture 2]. These labels are purely for identification and do not indicate any logical order or sequence in the trajectory. Your objective is to determine which picture represents the earlier step in the trajectory based on the given question description and careful analysis of the pictures.

2. **Relevance to the Task**: Both screenshots are directly related to either solving the question or finding relevant information to answer it. Do not assume that either image is irrelevant.

3. **Expert Trajectories**: These screenshots are taken from expert human trajectories that were executed to successfully solve the question. Trust that the screenshots reflect valid steps.

4. **Arbitrary Order**: The screenshots are presented in no particular order. Their arrangement does not indicate which step comes earlier in the trajectory.

Task:
{task}

Question:
Based on the two given images and the task, determine which image (or screenshot) represents an earlier step in the trajectory.

Follow the below guidelines to derive your answer:

1. **Understand the Task Context**:
Carefully read the task to identify the context and goal. This will help infer the logical progression of steps.

2. **Analyze Each Screenshot**:
Examine each screenshot for details like visible text, UI elements, and page structure. Identify clues about what action or information they represent in the process.

3. **Compare and Reason**:
Compare the screenshots using the question context and their visible elements. Consider dependencies or prerequisites to determine which screenshot or step must logically come first.

4. **Provide Your Answer**:
Clearly state which picture represents the **earlier step** in the trajectory, "Picture 1" or "Picture 2".

Output format:
Follow the format below for your response.

REASONING: Provide a concise step-by-step reasoning using the steps outlined above.

OUTPUT: Output your answer choice ("Picture 1" or "Picture 2"). Do not include any additional text here.

Figure 13

Future State Prediction (Instruction Prompting)

You are given an image showing the initial state of a form. This form requires the user to perform a series of actions before clicking the Submit button. Submitting the form navigates the user to a new window or page.

You are provided with two action plans (Plan A and Plan B) that describe the sequence of actions to interact with the form. Each plan is executed independently and in isolation, starting from the exact same initial state shown in the image. The two plans are not executed one after the other, and neither plan influences the result of the other.

Your task is to determine if the state of the form just before clicking Submit will be the same after following either plan.

Action Plans:
{plan_choices}

Question:
Based on the given image and the provided action plans, do both plans lead to the same state of the form just before clicking Submit when executed independently and in isolation, each starting from the same initial state of the form (i.e., shown in the image)?

Output format:
Only output "Yes" or "No".

Do not include any reasoning, explanation, or additional text.

Figure 14

Future State Prediction (Chain-of-Thought Prompting)

You are given an image showing the initial state of a form. This form requires the user to perform a series of actions before clicking the Submit button. Submitting the form navigates the user to a new window or page.

You are provided with two action plans (Plan A and Plan B) that describe the sequence of actions to interact with the form. Each plan is executed independently and in isolation, starting from the exact same initial state shown in the image. The two plans are not executed one after the other, and neither plan influences the result of the other.

Your task is to determine if the state of the form just before clicking Submit will be the same after following either plan.

Action Plans:
{plan_choices}

Question:
Based on the given image and the provided action plans, do both plans lead to the same state of the form just before clicking Submit when executed independently and in isolation, each starting from the same initial state of the form (i.e., shown in the image)?

Follow the steps provided next to arrive at your answer:

1. **Image Analysis**: Describe the relevant elements in the image, and carefully review the initial state of the form.
2. **Plan Evaluation**: Separately evaluate both the action plans step-by-step, and carefully determine how each step changes the state of the form.
3. **State Comparison**: Compare the final states of the form just before clicking Submit after executing Plan A and Plan B in isolation, each starting from the same initial state. Look for differences in the values and configurations of the form elements.
4. **Final Answer**: Based on your analysis, state whether both plans result in the same state of the form just before clicking Submit. Your final answer must be either "Yes" or "No".

Output format:
Your output should follow the below format:

REASONING: Be concise and to the point.

OUTPUT: "Yes" or "No". Do not include any additional text here.

Figure 15

Real Web Environment Study (Instruction Prompting)

You are given an image and a task that needs to be performed based on that image. The image represents what a user is currently seeing on their screen while attempting to solve the task. You are also provided with four choices of action plans (each represented as a set of actions) that outline potential steps to accomplish the task.

Your goal is to carefully analyze the image and evaluate the four action plans to decide which one most effectively helps the user complete the task.

Important Notes:
1. The current screenshot represents the user's current state or screen. As the user executes an action (e.g., clicking a button) from any plan, the screen or state may change. You must imagine these transformations and assess the feasibility of the remaining actions in each plan.

2. Some action plans may include invalid actions that cannot be executed based on the current or subsequent screens. It is your responsibility to identify and exclude such plans from being valid answer candidates.

3. The task is guaranteed to succeed within the next {plan_len} steps. If an action plan cannot complete the task within its steps, it is not a valid answer candidate.

4. Assume all plans are related to the task. However, some plans may include errors, inefficiencies, or invalid actions. Your responsibility is to evaluate each plan thoroughly to determine its effectiveness.

Task:
{task}

Plans (each represented as a series of actions):
{plan_text}

Question:
Based on the given image (the webpage screenshot), and the task,
which plan (A, B, C, D) would most effectively accomplish the task?

Output format:
Only output your choice of A, B, C, or D.

Do not include any reasoning, explanation, or additional text. Only provide your choice of plan (either A, B, C, D) in the response.

Figure 16

Real Web Environment Study (Chain-of-Thought Prompting)

You are given an image and a task that needs to be performed based on that image. The image represents what a user is currently seeing on their screen while attempting to solve the task. You are also provided with four choices of action plans (each represented as a set of actions) that outline potential steps to accomplish the task.

Your goal is to carefully analyze the image and evaluate the four action plans to decide which one most effectively helps the user complete the task.

Important Notes:
1. The current screenshot represents the user's current state or screen. As the user executes an action (e.g., clicking a button) from any plan, the screen or state may change. You must imagine these transformations and assess the feasibility of the remaining actions in each plan.

2. Some action plans may include invalid actions that cannot be executed based on the current or subsequent screens. It is your responsibility to identify and exclude such plans from being valid answer candidates.

3. The task is guaranteed to succeed within the next {plan_len} steps. If an action plan cannot complete the task within its steps, it is not a valid answer candidate.

4. Assume all plans are related to the task. However, some plans may include errors, inefficiencies, or invalid actions. It is your responsibility to evaluate them thoroughly.

Task:
{task}

Plans (each represented as a series of actions):
{plan_text}

Question:
Based on the given image (the webpage screenshot), and the task, which plan (A, B, C, D) would most effectively accomplish the task?

Follow these steps to derive your answer:

1. **Understand the Task**:
Read the task description to identify the goal and the likely steps required to accomplish it.

2. **Analyze the Screenshot**:
Examine the visible UI elements and details in the screenshot to determine the current state and what actions are possible.

3. **Evaluate the Plans**:
Assess each plan step-by-step by simulating the actions in sequence. Carefully check whether each action of a plan is valid based on the current or subsequent states to determine if that plan can complete the task within the given {plan_len} steps. Eliminate any plan with invalid, unnecessary, or inefficient actions.

4. **Select the Best Plan**:
Identify the plan that is valid, efficient, and effectively completes the task.

5. **Provide Your Answer**:
Briefly explain your reasoning and output your choice (A, B, C, or D).

Output format:
Follow the format below for your response.

REASONING: Provide a concise step-by-step reasoning using the steps outlined above.

OUTPUT: Output your answer choice ('A', 'B', 'C', or 'D').

Figure 17

Real Web Environment Study (Instruction Prompting) with Action History / Previous Action Input

You are given an image and a task that needs to be performed based on that image. The image represents what a user is currently seeing on their screen while attempting to solve the task. Along with this, you are given the exact set of previous actions the user has taken to reach the current state depicted in the image.

In addition, you are provided with four choices of action plans (each represented as a set of actions) that outline potential steps to accomplish the task. Your goal is to carefully analyze the image, review the previous actions, and evaluate the four action plans to determine which one most effectively helps the user complete the task.

Important Notes:
1. The current screenshot represents the user's current state or screen. As the user executes an action (e.g., clicking a button) from any plan, the screen or state may change. You must imagine these transformations and assess the feasibility of the remaining actions in each plan.

2. Some action plans may include invalid actions that cannot be executed based on the current or subsequent screens. It is your responsibility to identify and exclude such plans from being valid answer candidates.

3. The task is guaranteed to succeed within the next {plan_len} steps. If an action plan cannot complete the task within its steps, it is not a valid answer candidate.

4. Assume all plans are related to the task. However, some plans may include errors, inefficiencies, or invalid actions. Your responsibility is to evaluate each plan thoroughly to determine its effectiveness.

Task:
{task}

Previous actions:
{previous_actions_text}

Plans (each represented as a series of actions):
{plan_text}

Question:
Based on the given image (the webpage screenshot), the previous actions, and the task, which plan (A, B, C, D) would most effectively accomplish the task?

Output format:
Only output your choice of A, B, C, or D.

Do not include any reasoning, explanation, or additional text. Only provide your choice of plan (either A, B, C, D) in the response.

Figure 18

Edge Case Test Study (Instruction Prompting)

You are given an image and a task that needs to be performed based on that image. Additionally, you are given several possible action plans, each describing potential next steps to achieve the task. Your goal is to select the most effective plan from the options provided. Carefully analyze the image, and then evaluate each option to decide which one best completes the task.

Task:
{task}

Possible Action Plans:
{plan_text}

Question:
Based on the given image, and the given task, which action plan ({option_text}) would most effectively accomplish the task?

Output format:
Only output your choice of {option_text}.

Do not include any reasoning, explanation, or additional text.

Figure 19

only 2/19 models achieve an average performance of $\geq$50% under the Shuffling Perturbation setting, while models perform better in the Semantic Perturbation setting. In summary, we conclude that augmenting action history offers no significant benefit in improving VLMs' plan assessment capabilities.

I Additional Results and Details

In this section, we visualize outputs and include tables that we could not include in the main paper due to space constraints. Specifically, we present more results for: (1) VLMs' spurious preferences in Temporal Ordering task (Fig. 24), (2) Unreasonable quality of GUI-specific models (Fig. 25), (3) Samples with erroneous responses for Future State Prediction (Fig. 27) and Real-Web Plan Assessment Analysis (Fig. 26), (4) Comprehensive tables detailing results for each individual task—Task 1, Task 2, and Task 3 of the Edge Case Test (see Tab. 13), as well as Task 1, Task 2, and Task 3 of the Future State Consistency test (see Tab. 11, Tab. 12), (5) Dataset Statistics summarizing number of test examples for every configuration of our paper (see Tab. 10).

J Would VLMs perform perfectly if Perception error did not exist?

In the error case study for the edge case test (Fig. 4), we observe that "Perception Error," where VLMs misinterpret certain visual details in the image, emerges as a bottleneck for plan identification. While addressing this issue is beyond the scope of our work, we explore how VLM performance changes when provided with accurate ground truth captions describing the form. Specifically, we aim to assess whether VLMs can reason about plans even after getting an accurate and complete textual description of the form screenshot. For simplicity, we conduct this test using Task 1 from the Edge Case Test Study and evaluate five VLMs. We compare performance across three configurations – Image Only, Caption Only, and Image + Captions. Our results in Tab. 14 reveal two key findings: (1) Although adding ground truth captions improves performance over the Image-Only configuration, the selected models still fall significantly short of reasonable performance, and (2) while ground truth captions provide slight improvement, the Image + Caption configuration is either worse than Caption-Only or offers no noticeable advantage. Thus, highlighting critical limitations in the reasoning and

Edge Case Test Study (Chain-of-Thought Prompting)

You are given an image and a task that needs to be performed based on that image. Additionally, you are given several possible action plans, each describing potential next steps to achieve the task. Your goal is to select the most effective plan from the options provided. Carefully analyze the image, and think step by step to evaluate each option to decide which one best completes the task.

Task:
{task}

Possible Action Plans:
{plan_text}

Question:
Based on the given image, and the given task, please follow these steps:

1. **Image and Task Analysis:** Describe relevant elements in the image and clarify the requirements of the task.
2. **Option Evaluation:** Go through each action plan ({option_text}) one by one, and describe why it would or would not help achieve the task. Mention any advantages or limitations of each option based on the image.
3. **Final Choice:** Based on your analysis, select the option ({option_text}) that best accomplishes the task.

Output Instructions:
Your output should follow the below format:

REASONING: Be concise and to the point.

OUTPUT CHOICE: An uppercase letter

Figure 20

planning capabilities of contemporary VLMs.

K Additional Related Work: Importance of Small Models for Web Planning

While several recent studies have employed proprietary models for web agents, our work emphasizes the use of open-source models, many of which—despite being small—have been widely adopted in the web planning literature. For instance, prior works (Pahuja et al., 2025; Chen et al., 2024; Cheng et al., 2024) utilize models such as Qwen-VL and Minicpm, while others leverage models like Phi-3.5v, Ferret-UI, UIX (LLaVA-based), and CogAgent for GUI-specific tasks (Xu et al., 2025; Hong et al., 2024; Cheng et al., 2024). Unlike models that solely generate actions, many of these approaches train models to produce thoughts or reasoning steps prior to action execution, thereby implicitly requiring planning capabilities. Notably, Qwen2-VL models have been explicitly trained with planning and reasoning datasets to support such behavior.

Given the high costs, proprietary constraints, and privacy concerns associated with large-scale commercial models, the broader research community is increasingly gravitating toward smaller, open-access alternatives. Our study aligns with this direction by providing timely insights into the intrinsic planning capabilities of vision-language models (VLMs) that are already seeing practical use in web-based tasks.

Although resource limitations prevented us from conducting the full experiment suite on proprietary models, we conducted additional trials using GPT-4o-mini to demonstrate the broader potential of

Llama as a judge for Temporal Ordering

You are acting as a judge. The statement below describes in text which image comes earlier in a sequence. Parse it carefully and decide which picture comes earlier.
Respond strictly with either **"Picture 1"** or **"Picture 2"** and no additional text:

Statement:
{output}

Figure 21

Llama as a judge for Future State Prediction

You are provided with a response to the following question:

"Do both plans (Plan A and Plan B) result in the same final state of the form (i.e., just before clicking Submit)?"

Your task is to analyze the response and determine the answer based on the following criteria:

1. Output "yes" if the response explicitly states that both Plan A and Plan B achieve the same final state.
2. Output "no" if the response explicitly states that Plan A and Plan B do not achieve the same final state.
3. Output "unclear" if the response does not clearly state whether Plan A and Plan B achieve the same final state.

Respond with only the answer: "yes", "no", or "unclear". Provide no additional text.

****Response:****
{response}

****Answer:****

Figure 22

Llama as a judge for Plan Assessment (both Real-World and Edge Case)

You are acting as a judge. The statement below describes in text which Action Plan Option best accomplishes the task. Parse it carefully.

Your task is to analyze the statement and determine the answer based on the following criteria:

1. Output "A" if the statement states "Plan A" or "Option A" or anything related.
2. Output "B" if the statement states "Plan B" or "Option B" or anything related.
3. Output "C" if the statement states "Plan C" or "Option C" or anything related.
4. Output "D" if the statement states "Plan D" or "Option D" or anything related.
5. Output "unclear" if the statement does not state which Action or Option best accomplishes the task.

Respond with only the answer: "A", "B", "C", "D", or "unclear". Provide no additional text.

Statement: {response}

Answer:

Figure 23

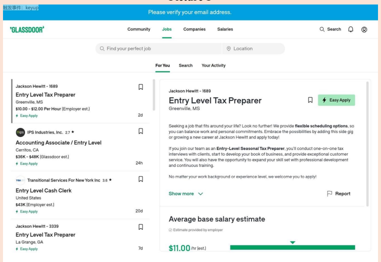
our approach. In this work, we focus on two core tasks—Temporal Ordering and Plan Assessment for real-world web environments. For temporal prediction, we utilize the GUIAct dataset and evaluate a sample of 500 questions from the full set. For Plan Assessment, we assess model behavior under the Shuffle Perturbation setting.

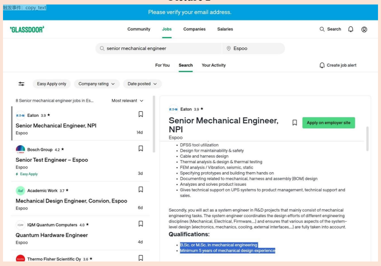
L Future Work

As part of our future work to mitigate the limitations, we aim to incorporate the fine-grained skills and insights highlighted in this work into contemporary VLMs. Specifically, each of our tests offers guidance on the types of tasks and data that could serve as essential inductive biases and should be included during the pre-training phase of VLMs. Our main focus is on injecting these basic skills into smaller agents (less than or equal to 7B) rather than larger ones. This emphasis stems from the belief that for agentic workflows to achieve widespread adoption, smaller models are more practical, as they can be deployed on-device, ensuring user privacy (since agents will handle end user's day-to-day tasks) and enabling faster inference. Therefore, we call for a collective community effort to address the fundamental performance shortcomings of such models, as revealed in our study, through novel task designs and training.

Task: Find the qualifications for Senior Mechanical Engineer jobs in Espoo, Finland.

Picture 1 | **Picture 2**



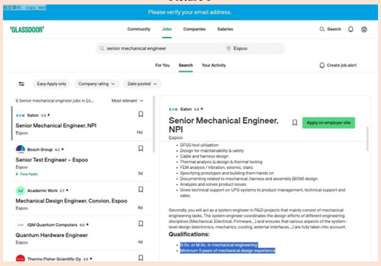


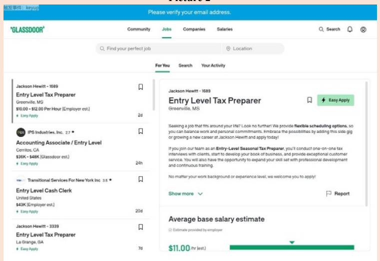
Reasoning: To determine the earlier step in the trajectory, we need to understand the logical sequence of actions required to find the qualifications for Senior Mechanical Engineer jobs in Espoo, Finland. The first step would likely involve searching for job listings, which is shown in Picture 1. Once job listings are found, the next step would be to review the qualifications for those jobs, which is depicted in Picture 2.\n\nPicture 1 shows a job search interface with various job listings, indicating that the user is in the process of searching for jobs. Picture 2 shows a detailed job listing for a Senior Mechanical Engineer position, suggesting that the user has found a relevant job and is now reviewing the qualifications for that specific position.\n\nTherefore, the earlier step in the trajectory is finding job listings, which is shown in Picture 1.\n\n

Output: Picture 1

Task: Find the qualifications for Senior Mechanical Engineer jobs in Espoo, Finland.

Picture 1 | **Picture 2**





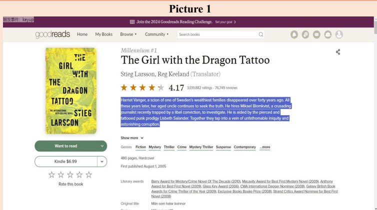
Reasoning: To determine the earlier step in the trajectory, we need to understand the logical sequence of actions required to answer the question about the qualifications for Senior Mechanical Engineer jobs in Espoo, Finland. The first step would likely involve searching for relevant job listings, which is depicted in Picture 1. Once the job listings are found, the next step would be to review the qualifications listed in the job descriptions, which is shown in Picture 2.\n\nPicture 1 shows the search results for Senior Mechanical Engineer jobs in Espoo, Finland, which is the initial step in finding relevant job information. Picture 2 displays the detailed qualifications for one of the job listings, which is the subsequent step after identifying the job.\n\nTherefore, the earlier step in the trajectory is finding the job listings, which is represented by Picture 1.\n\n

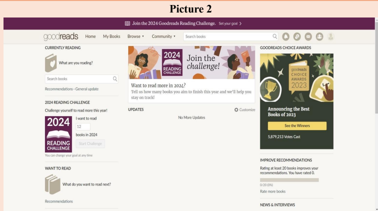
Output: Picture 1

(a) Example 1

Task: Find me some information about the book The Girl with the Dragon Tattoo by Stieg Larsson.

Picture 1 | **Picture 2**



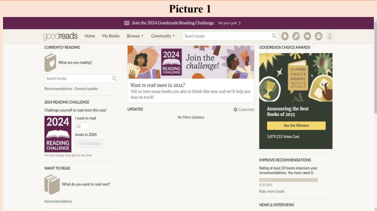


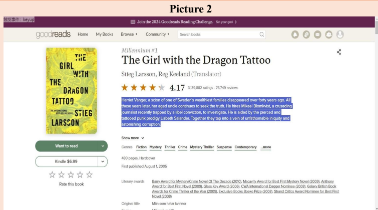
Reasoning: The first image provides detailed information about the book \"The Girl with the Dragon Tattoo\" by Stieg Larsson, including the book's cover, title, author, and a brief synopsis. This suggests that the user is looking for specific details about the book, such as its content and background.\n\nThe second image, on the other hand, shows the interface of the Goodreads website, specifically the \"2024 Reading Challenge\" section. This indicates that the user is engaging with the website to set a reading goal for the year 2024, which is unrelated to the specific details of the book \"The Girl with the Dragon Tattoo.\"\n\nGiven the context of the question, which is to find information about the book, the first image logically represents an earlier step in the trajectory because it provides relevant information about the book itself. The second image is more about the user's engagement with the Goodreads platform and does not contribute to finding information about the book.\n\nTherefore, the earlier step in the trajectory is represented by \"Picture 1\".\n\n

Output: Picture 1

Task: Find me some information about the book The Girl with the Dragon Tattoo by Stieg Larsson.

Picture 1 | **Picture 2**



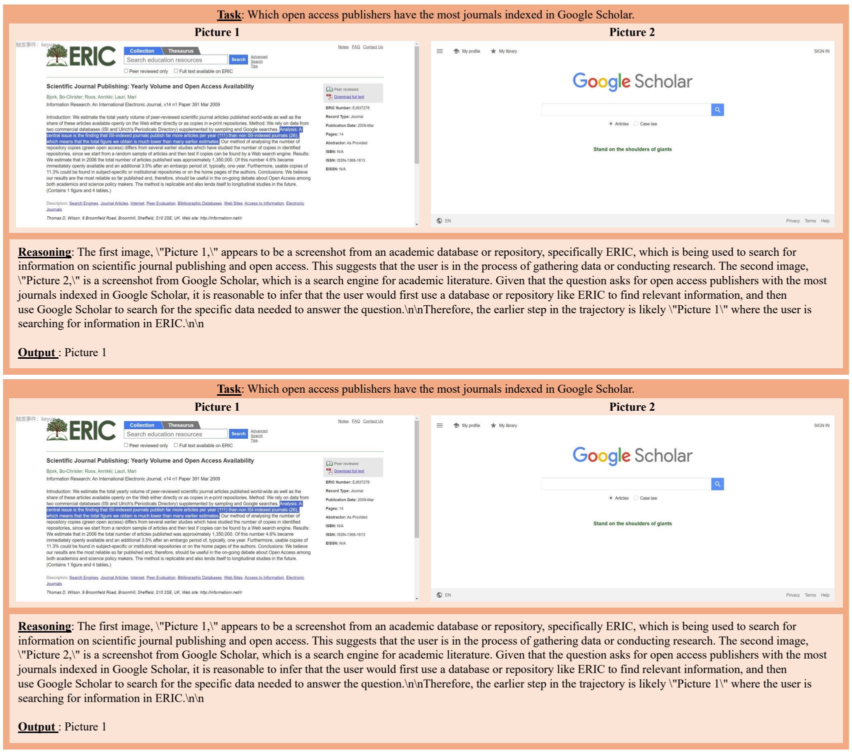


Reasoning: To find information about the book \"The Girl with the Dragon Tattoo\" by Stieg Larsson, one would likely start by searching for the book on the Goodreads website. Picture 1 shows the Goodreads homepage with options to search for books, which aligns with the initial step of searching for the book. Picture 2 shows a detailed book page, which suggests that after finding the book, one would click on it to get more information. Therefore, the earlier step in the trajectory would be searching for the book on the homepage.\n\n

Output: Picture 1

(b) Example 2

Figure 24



(c) Example 3

Figure 24: In the main paper, we find that VLMs are biased towards answering ’Picture 1’ for the temporal ordering task. Specifically, the mean performance difference across all models, between cases where “Picture 1” is the correct answer and cases where “Picture 2” is the correct answer, is 46.64% for MCQ and 41.66% for non-MCQ versoins. Here, we show qualitative results how VLMs tend to rationalize their answer of “Picture 1” with untenable reasons

Main	Shuffle Perturb.			Semantic Perturb.		
	INS	COT	Avg	INS	COT	Avg
Qwen2-VL 2B	41.88	23.01	32.45	52.45	28.67	40.56
Qwen2-VL 7B	56.22	48.3	52.26	87.92	71.69	79.81
Qwen2-VL 72B	79.24	69.05	74.14	95.09	90.94	93.02
Minicpm	55.47	36.22	45.84	83.01	63.01	73.01
Minicpm-o	49.05	39.24	44.14	73.0	70.56	71.78
Phi 3.5	27.92	33.2	30.56	56.6	55.09	55.84
IDEFICS3	43.39	33.58	38.48	80.75	55.47	68.11
DeepSeek VL	28.67	20.0	24.34	38.87	25.66	32.27
Llava OV 0.5B	24.52	8.3	16.41	27.02	8.3	17.66
Llava OV 7B	56.6	41.5	49.05	89.05	75.09	82.07
InternVL2-MPO	46.41	30.56	38.48	90.56	71.32	80.94
Glm	28.3	39.62	33.96	64.52	59.62	62.07
Llava 1.6 - 7B	26.41	30.18	28.30	38.86	46.59	42.73
Llava 1.6 - 13B	30.94	29.43	30.19	52.45	48.3	50.38
Llava 1.6 - 34B	41.50	41.50	41.50	87.16	70.45	78.81
Ferret-UI-Llama	23.1	26.03	24.57	26.03	23.77	24.90
Ferret-UI-Gemma	12.07	21.13	16.60	9.88	21.21	15.55
UIX-Llava	48.3	33.2	40.75	67.54	41.88	54.71
CogAgent	23.86	22.34	23.10	23.77	17.73	20.75

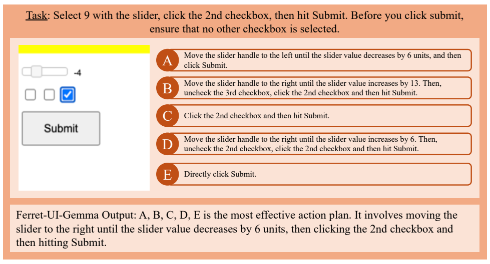
Table 9: Evaluations of 19 VLMs on Plan Selection in Real-Web Environment. In this particular experiment, we augment VLM input with the history of its previous actions. While we anticipated an improvement in performance after using information of previous actions, we found that VLMs reaped no substantial benefit from the aforementioned.

Test Name	Configuration	Number of Examples
Temporal Ordering - MM-M2W	Total	**2018**
	Order1	1009
	Order2	1009
Temporal Ordering - GUIAct	Total	**4742**
	Order1	2371
	Order2	2371
Future State Prediction - Long Horizon	Total	**3838**
	Task 1	960
	Task 2	1816
	Task 3	1062
Future State Prediction - Short Horizon	Total	**3358**
	Task 1	960
	Task 2	1336
	Task 3	1062
Plan Selection - Edge Case Scenario	Total	**1530**
	Task 1	531
	Task 2	500
	Task 3	499
Plan Selection - Real Web Environment	Total	**435**

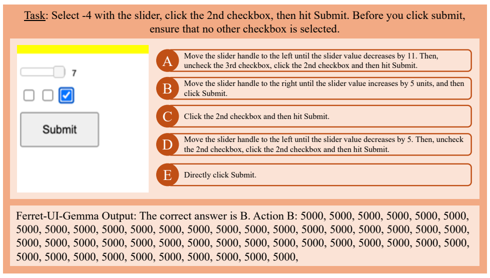
Table 10: Dataset Statistics: Number of examples for each test and configuration.

Main	Task 1		Task 2		Task 3		Avg.
	INS	COT	INS	COT	INS	COT	
Qwen2-VL 2B	73.43	63.64	43.98	49.32	49.62	47.08	54.51
Qwen2-VL 7B	99.27	90.48	50.81	81.27	50.00	57.25	71.51
Qwen2-VL 72B	97.50	97.91	97.66	98.37	99.71	95.76	97.82
Minicpm	52.39	66.67	50.99	64.93	50.00	56.04	56.84
Minicpm-o	70.10	82.25	40.76	67.47	60.64	55.57	62.80
Phi 3.5	87.50	81.77	61.19	62.42	53.86	56.68	67.24
IDEFICS3	53.22	96.56	0.5	87.34	50.28	54.80	57.12
DeepSeek VL	58.85	79.89	44.35	53.02	50.65	39.64	54.40
Llava OV 0.5B	50.00	43.95	50.00	40.78	52.35	48.11	47.53
Llava OV 7B	55.83	87.18	52.14	76.47	60.16	65.72	66.25
InternVL2-MPO	62.91	96.77	20.96	92.04	28.53	72.10	62.22
Glm	50.31	68.64	50.23	57.07	63.46	58.28	58.00
Llava 1.6 - 7B	92.08	82.29	58.91	78.08	44.82	32.16	64.72
Llava 1.6 - 13B	50.00	76.14	50.00	63.46	50.00	47.87	56.25
Llava 1.6 - 34B	91.14	91.87	83.03	81.94	65.91	55.36	78.21
Ferret-UI-Llama	50.00	50.10	50.00	49.88	40.29	50.00	48.38
Ferret-UI-Gemma	69.00	5.72	44.73	2.75	64.23	10.07	32.75
UIX-Qwen2	60.52	70.20	59.94	63.00	32.01	54.80	56.75
CogAgent	50.00	30.93	50.00	32.06	50.00	43.97	42.83

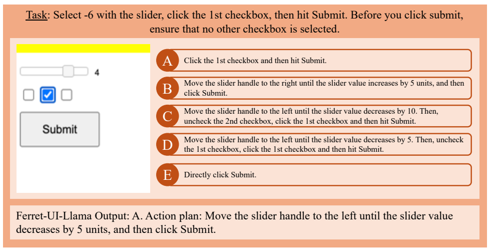
Table 11: Task-wise results for Future State Prediction in short horizon action scenarios



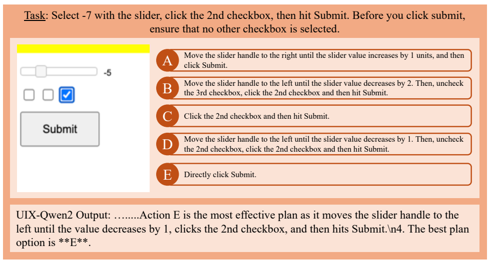
(a) GUI Model Outputs – Irrelevant/Gibberish



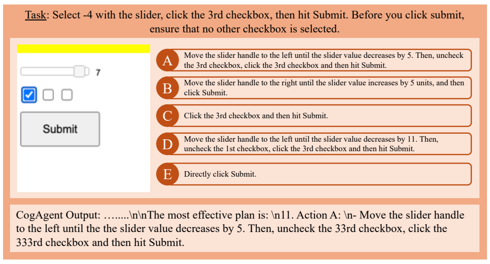
(b) GUI Model Outputs – Irrelevant/Gibberish



(c) GUI Model Outputs – Wrong option description



(d) GUI Model Outputs – Wrong option description



(e) GUI Model Outputs – Wrong option description

Figure 25: In the main paper, we discuss how GUI models have worse performances than general VLMs. We now demonstrate qualitative outputs eliciting deterioration in GUI models' general reasoning and instruction following capabilities. More specifically, fig (a), (b) represent cases where model starts producing gibberish content and does not choose an option from the given choices. Fig (c), (d), (e) has examples where model did choose an option but gave an incorrect description of what the option actually is in the question.

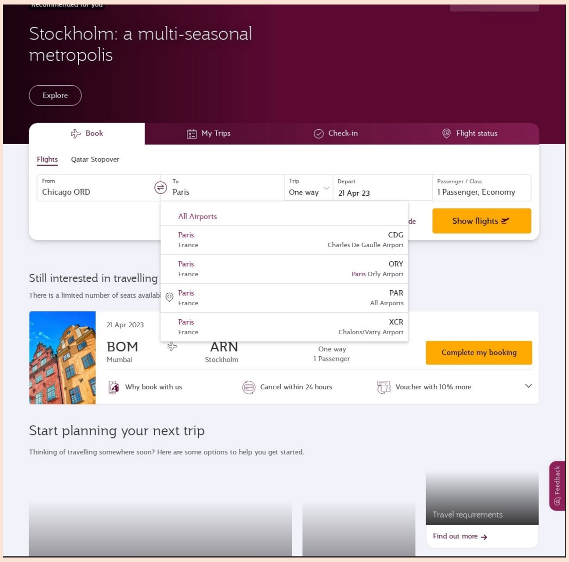
Main	Task 1		Task 2		Task 3		Avg.
	INS	COT	INS	COT	INS	COT	
Qwen2-VL 2B	52.60	51.56	48.95	48.12	49.71	48.39	49.89
Qwen2-VL 7B	84.27	55.75	68.07	38.5	50.0	50.90	57.91
Qwen2-VL 72B	100	57.70	45.81	55.67	84.83	88.32	55.55
Minicpm	50.0	48.00	50.00	2.29	50.00	56.42	42.78
Minicpm-o	50.41	54.08	86.89	40.27	36.15	52.14	53.32
Phi 3.5	74.06	54.27	11.56	34.14	49.71	53.01	46.13
IDEFICS3	50.83	51.97	49.39	34.63	50.00	64.87	50.28
DeepSeek VL	50.00	51.56	30.34	45.26	49.43	40.58	44.53
Llava OV 0.5B	50.00	42.81	50.00	40.91	47.83	48.30	46.64
Llava OV 7B	50.00	51.66	51.10	44.49	49.24	60.73	51.20
InternVL2-MPO	75.62	65.52	39.20	46.21	48.77	67.48	57.13
Glm	50.62	55.20	50.00	43.85	48.68	52.30	50.11
Llava 1.6 - 7B	58.22	56.04	22.19	41.18	41.43	48.68	44.62
Llava 1.6 - 13B	50.00	55.41	50.00	34.58	50.0	47.12	47.85
Llava 1.6 - 34B	72.08	65.729	22.46	48.84	43.97	46.79	49.98
Ferret-UI-Llama	100	49.79	50.0	49.77	49.23	50.00	58.13
Ferret-UI-Gemma	52.36	4.89	44.80	6.55	49.92	6.40	27.49
UIX-Qwen2	51.14	51.04	50.00	44.76	19.20	44.35	43.41
CogAgent	50.00	48.54	50.00	38.21	50.00	42.93	46.61

Table 12: Task-wise results for Future State Prediction in long horizon action scenarios

Main	Task 1		Task 2		Task 3	
	INS	COT	INS	COT	INS	COT
Qwen2-VL 2B	27.11	28.92	26.66	32.20	59.71	45.29
Qwen2-VL 7B	13.74	11.86	67.20	62.40	67.73	69.27
Qwen2-VL 72B	95.29	88.51	83.40	81.40	87.57	58.83
Minicpm	1.6	10.54	42.60	25.00	57.11	65.93
Minicpm-o	26.36	20.15	49.60	32.60	67.33	59.31
Phi 3.5	21.84	12.05	26.00	16.00	50.70	50.50
IDEFICS3	27.11	43.69	59.20	64.00	52.30	59.91
DeepSeek VL	0.9	12.24	13.80	38.60	40.28	33.66
Llava OV 0.5B	14.87	20.52	28.60	22.20	14.87	22.24
Llava OV 7B	19.39	13.93	65.20	53.60	28.05	47.69
InternVL2-MPO	45.38	24.14	75.60	66.20	52.10	57.11
Glm	25.80	28.86	54.00	42.60	60.52	55.22
Llava 1.6 - 7B	26.55	31.45	33.40	41.60	28.05	47.69
Llava 1.6 - 13B	26.36	32.01	24.40	33.60	27.51	30.26
Llava 1.6 - 34B	14.90	11.48	35.60	29.20	70.54	48.49
Ferret-UI-Llama	28.06	23.6	18.40	23.60	16.03	20.64
Ferret-UI-Gemma	10.73	12.42	8.00	12.80	16.43	14.62
UIX-Qwen2	0.3	10.73	37.00	8.00	30.66	24.64
CogAgent	16.57	22.22	41.40	21.80	21.04	19.23

Table 13: Task-wise results for Plan Selection – Edge case Test

Task:Show me the list of one-way flights today (April 17) from Chicago to Paris



Plan A

1. Click on the Show flights button to view the available flight options.",
2. Click on the Continue button to proceed with the booking process.",
3. Select 17 April 2023 from the available grid cell for the departure date.",
4. Click on the Paris France CDG Charles De Gaulle Airport menu item to select the departure location.",
5. Choose the One way option for the type of trip."

Plan B

1. Click on the Add to Cart link to include the item in your shopping cart.",
2. Click on the Buy a copy link to proceed with purchasing options.",
3. Click on the Gorillaz link to explore more about Gorillaz.",
4. Click on the Gorillaz Gorillaz Artist link to view the artist's page."

Plan C

1. Click on the Show flights button to view the available flight options.",
2. Click on the Continue button to proceed with the booking process.",
3. Select 17 April 2023 from the available grid cell for the departure date.",
4. Choose the One way option for the type of trip.",
5. Click on the Paris France CDG Charles De Gaulle Airport menu item to select the departure location."

Plan D

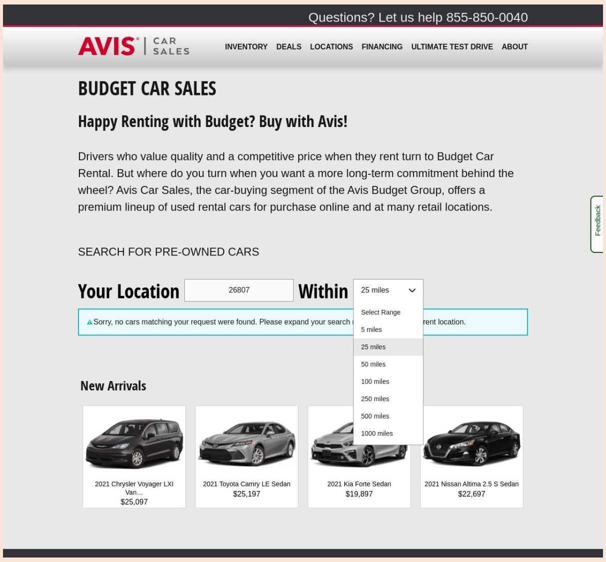
1. Click on the Continue button to proceed with the booking process.",
2. Select 17 April 2023 from the available grid cell for the departure date.",
3. Choose the One way option for the type of trip.",
4. Click on the Paris France CDG Charles De Gaulle Airport menu item to select the departure location.",
5. Click on the Show flights button to view the available flight options."

Reasoning: "REASONING: \n1. The task requires showing a list of one-way flights today (April 17) from Chicago to Paris.\n2. The screenshot shows a booking form with fields for departure and destination airports, trip type, and departure date.\n3. Plan A starts by clicking the \"Show flights\" button, which is not the correct first step since the departure date needs to be set first.\n4. Plan B selects the destination airport first, which is incorrect since the departure date needs to be set first.\n5. Plan C starts by clicking the \"Show flights\" button, which is incorrect as the departure date needs to be set first.\n6. Plan D starts by clicking the \"Continue\" button, which is not the correct first step since the departure date needs to be set first.\n7. None of the plans correctly set the departure date as the first step, which is necessary to complete the task.\n\nOUTPUT: None of the plans (A, B, C, D) effectively accomplish the task.\n"

Output Choice: None of the plans (A, B, C, D) effectively accomplish the task.\n

(a) Shuffling Perturbation: Example 1

Task: Find a Buick for sale within 250 miles from 26807.



Plan A

1. Click on the image to view a larger picture of the vehicle.
2. Select the Buick checkbox to filter the vehicle options.
3. Click on the information icon for more details.
4. Click on the View Vehicles button to see available vehicles.
5. Click on the 250 miles from the available options.

Plan B

1. Click on the 250 miles from the available options.
2. Click on the View Vehicles button to see available vehicles.
3. Click on the information icon for more details.
4. Select the Buick checkbox to filter the vehicle options.
5. Click on the image to view a larger picture of the vehicle.

Plan C

1. Click on the image to view a larger picture of the vehicle.
2. Select the Buick checkbox to filter the vehicle options.
3. Click on the information icon for more details.
4. Click on the 250 miles from the available options.
5. Click on the View Vehicles button to see available vehicles.

Plan D

1. Select the Buick checkbox to filter the vehicle options.
2. Click on the information icon for more details.
3. Click on the View Vehicles button to see available vehicles.
4. Click on the 250 miles from the available options.
5. Click on the image to view a larger picture of the vehicle.

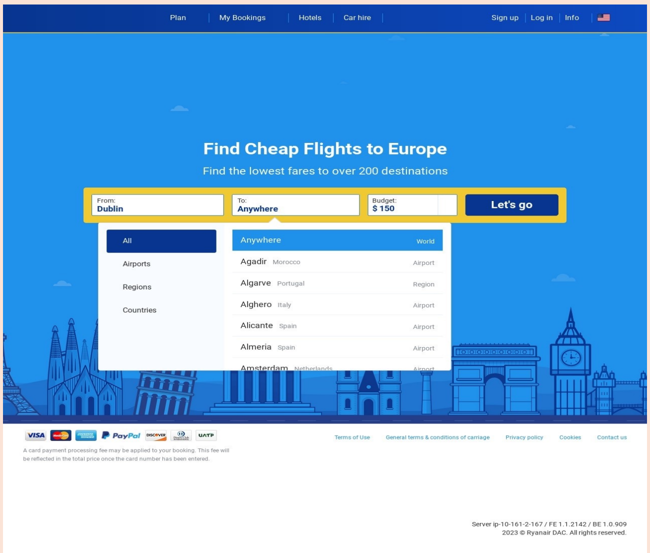
Reasoning: \n1. The task is to find a Buick for sale within 250 miles from 26807.\n2. The screenshot shows a search interface with options to enter a location and select a range.\n3. Plan A starts with an irrelevant action (viewing a larger picture of the vehicle) and does not include selecting the Buick checkbox.\n4. Plan B also starts with an irrelevant action (viewing a larger picture of the vehicle) and does not include selecting the Buick checkbox.\n5. Plan C starts with an irrelevant action (viewing a larger picture of the vehicle) and does not include selecting the Buick checkbox.\n6. Plan D includes selecting the Buick checkbox, viewing vehicle details, and selecting the 250-mile range, which are all relevant actions to complete the task.

Output: D\n

(b) Shuffling Perturbation: Example 2

Figure 26

Task: Use the FareFinder to find flights for $150 or less from Dublin to anywhere.



Plan A

1. Click the Let's go button to continue with the booking process.
2. Select the price indicator showing $150 for further details.
3. Click on the 'Anywhere' icon to proceed with the selection.
4. Click on the 'To' section to explore destination options.

Plan B

1. Select the price indicator showing $150 for further details.
2. Click the Let's go button to continue with the booking process.
3. Click on the 'Anywhere' icon to proceed with the selection.
4. Click on the 'To' section to explore destination options.

Plan C

1. Click the Let's go button to continue with the booking process.
2. Select the price indicator showing $150 for further details.
3. Click on the 'To' section to explore destination options.
4. Click on the 'Anywhere' icon to proceed with the selection.

Plan D

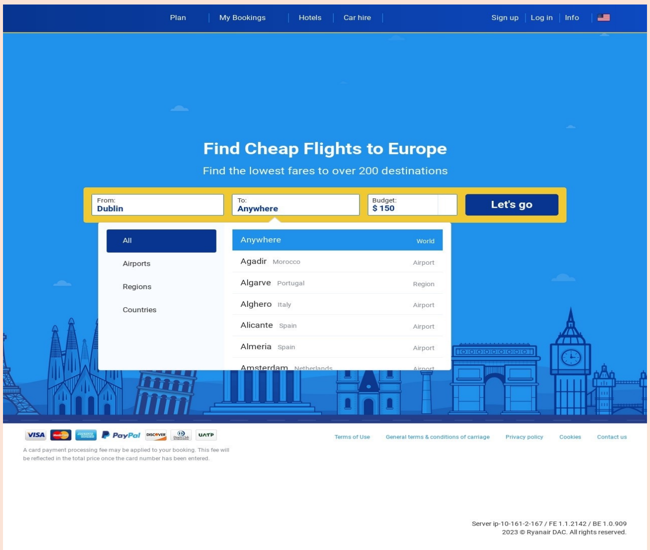
1. Click on the 'To' section to explore destination options.
2. Click on the 'Anywhere' icon to proceed with the selection.
3. Select the price indicator showing $150 for further details.
4. Click the Let's go button to continue with the booking process.

Reasoning:\n1. The task is to find flights for $150 or less from Dublin to anywhere using the FareFinder.\n2. The screenshot shows the FareFinder interface with fields for \"From,\" \"To,\" and \"Budget,\" and a \"Let's go\" button.\n3. Plan A starts by clicking the \"Let's go\" button, which is not the correct first step since the budget needs to be set first.\n4. Plan B starts by selecting the price indicator showing $150, which is the correct first step, followed by clicking the \"Let's go\" button, which is the correct second step.\n5. Plan C starts by clicking the \"Let's go\" button, which is incorrect as the budget needs to be set first.\n6. Plan D starts by clicking on the \"To\" section, which is incorrect as the budget needs to be set first.

Output: B\n

(c) Shuffling Perturbation: Example 3

Task: Use the FareFinder to find flights for $150 or less from Dublin to anywhere.



Plan A

1. Click the Let's go button to continue with the booking process.
2. Select the price indicator showing $150 for further details.
3. Click on the HAPPY HOUR BURGER link for the event on Thursday, April 13, 2023.
4. Click on the 'To' section to explore destination options.

Plan B

1. Click on the Flight status link to check the details.
2. Click the Let's go button to continue with the booking process.
3. Click on the 'Anywhere' icon to proceed with the selection.
4. Click on the 'To' section to explore destination options.

Plan C

1. Click the Let's go button to continue with the booking process.
2. Select the price indicator showing $150 for further details.
3. In the ZIP text box, type \"11101\" as the postal code.
4. Click on the 'Anywhere' icon to proceed with the selection.

Plan D

1. Click on the 'To' section to explore destination options.
2. Click on the 'Anywhere' icon to proceed with the selection.
3. Select the price indicator showing $150 for further details.
4. Click the Let's go button to continue with the booking process.

Reasoning: \n1. The task is to find flights for $150 or less from Dublin to anywhere using the FareFinder.\n2. The previous actions show that the user has already entered Dublin as the departure location and is currently selecting the destination.\n3. In the screenshot, the user has selected \"Anywhere\" as the destination and set the budget to $150.\n4. Plan A involves checking flight status and selecting a price indicator, which is not relevant to the task.\n5. Plan B involves typing an unrelated location and clicking the Let's go button, which is not relevant to the task.\n6. Plan C involves typing a postal code, which is not relevant to the task.\n7. Plan D involves selecting \"Anywhere\" as the destination and then selecting the price indicator showing $150, which aligns with the task.\n8. Plan D is the only plan that directly addresses the task and can be completed within the given 4 steps.

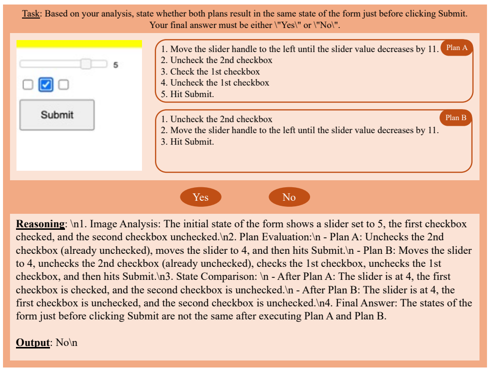
Output: D\n

(d) Semantic Perturbation: Example 4

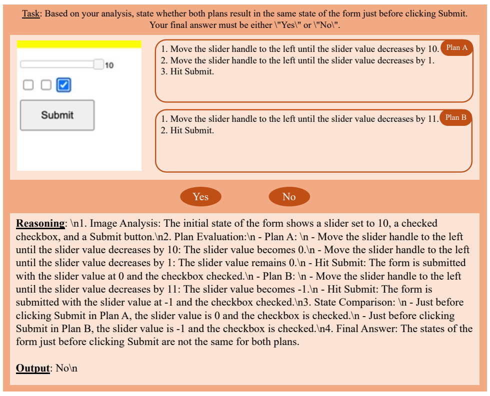
Figure 26: Qualitative results for Plan Assessment - Real web scenario. The above figure shows outputs from Qwen2-VL 72B. Example 1, 2, 3 includes results for Shuffling Perturbation whereas Example 4 is the result for Semantic Perturbation.

	Img Only			Img + Cap			Cap Only		
	INS	CoT	Avg	INS	CoT	Avg	INS	CoT	Avg
Minicpm-o	26.36	20.15	23.26	33.90	20.53	27.21	34.46	20.90	27.68
Qwen2-VL-7B	13.74	11.86	12.81	15.25	3.58	9.42	13.94	9.04	11.49
Phi 3.5	21.84	12.05	16.95	32.77	25.61	29.19	32.96	27.87	30.41
Glm	25.80	28.86	27.33	23.35	22.03	22.69	22.22	24.29	23.26
UIX-Qwen2	0.3	10.73	5.56	4.52	9.04	6.78	3.77	9.04	6.40

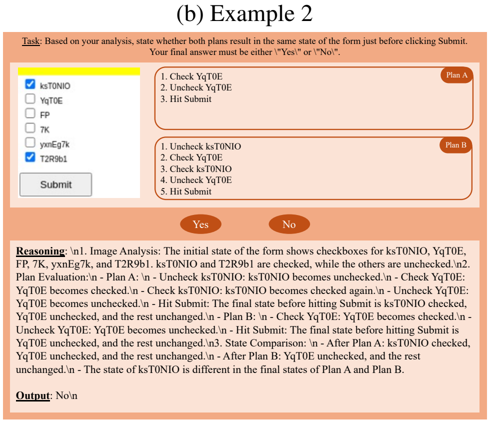
Table 14: Performance comparison under different configurations (Img Only, Img + Cap, and Cap Only) across INS, CoT, and Avg metrics.



(a) Example 1



(b) Example 2



(c) Example 3

Figure 27: Qualitative results for Future State Prediction.
The above figure shows outputs from Qwen2-VL 72B.