HALO: Towards Systematic Evaluation and Online Detection of Hallucination in Multimodal Agents

Anonymous ACL submission

Abstract

Multimodal large language models (MLLMs) are increasingly evolving into multimodal agents capable of perceiving visual environments, invoking tools, and performing multi-step interactions. However, existing hallucination evaluations primarily focus on static visionlanguage tasks, providing only a limited view of hallucination behaviors in such agentic settings. In practice, hallucinations may arise not only from incorrect visual grounding but also from failures in reasoning, memory, or action execution, and can propagate across interaction trajectories. To address this gap, we introduce HALO, a benchmark for evaluating hallucination in multimodal agents from both capability-level and trajectory-level perspectives. HALO includes HALO-Static, a capability-level benchmark covering 12 representative tasks with 3,517 QA pairs, and HALO-Dynamic, a step-level hallucination detection and classification task evaluated on agent trajectories collected from VisualWebArena. Experiments on a wide range of LVLMs show that current models still exhibit substantial hallucination across agent capabilities and struggle to detect hallucinations in interaction trajectories reliably.

1 Introduction

Multimodal large language models (MLLMs) (Bai et al., 2025b; Liu et al., 2023; Maaz et al., 2024) have attracted growing attention in recent years due to their strong potential for real-world applications. Despite rapid progress, these models still suffer from severe hallucination issues, referring to the misalignment between factual visual content and the generated textual descriptions, which poses a significant challenge to their reliable deployment (Jing et al., 2024; Lovenia et al., 2023). Meanwhile, MLLMs are rapidly evolving into multimodal agents capable of perceiving visual scenes, retrieving information, invoking tools, executing actions, and engaging in multi-turn interactions. This shift raises an important question: *do multimodal agents exhibit new forms of hallucination beyond those observed in static vision-language settings?*

Existing hallucination evaluations primarily focus on static vision-language settings, where models generate captions or answers grounded in a single image (Li et al., 2023; Rohrbach et al., 2018). These benchmarks typically measure whether generated descriptions align with visual evidence, providing only a vision-centric view of hallucination. However, multimodal agents operate in substantially more complex environments involving sequential decision-making, tool usage, environment feedback, and multi-step reasoning. In such settings, hallucinations may arise not only from incorrect visual grounding but also from failures in core agent capabilities such as reasoning, memory, or action selection. Moreover, real agents execute long trajectories where hallucinations may emerge at intermediate steps and propagate across interactions. As illustrated in Figure 1, evaluation protocols designed for static MLLMs outputs are insufficient to characterize hallucination in multimodal agents.

To address these limitations, we introduce HALO, a benchmark for evaluating hallucination in multimodal agents. HALO consists of two complementary settings. First, *HALO-Static* evaluates capability-level hallucinations across core agent abilities, including perception, reasoning, memory, execution, and communication. The benchmark is derived from existing multimodal agent tasks (Koh et al., 2024; Liu et al.) and covers 12 representative task categories. Each instance is formulated as a multiple-choice QA problem, resulting in a dataset of 3,517 QA pairs. However, real multimodal agents operate through long interaction trajectories, where errors may arise at intermediate steps and propagate across decisions.

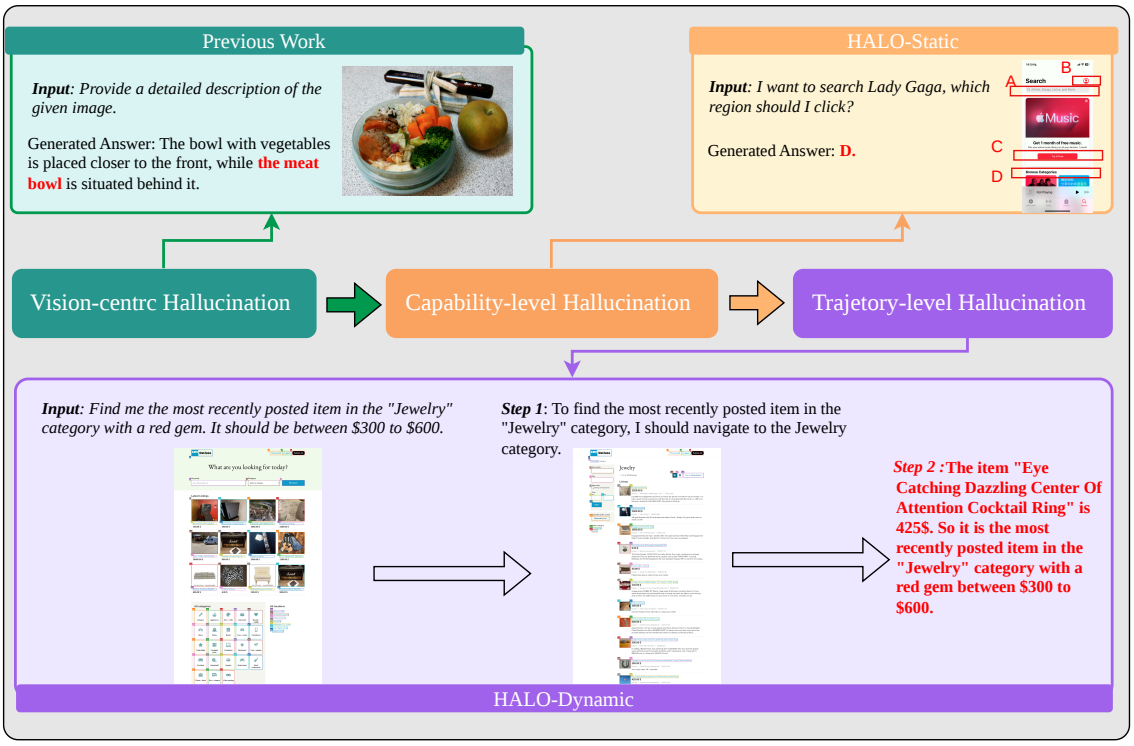


Figure 1: Comparison of hallucination evaluation across different abstraction levels. Previous work primarily studies vision-centric hallucination, focusing on static inconsistencies between visual inputs and generated outputs. HALO extends this perspective by introducing capability-level evaluation (HALO-Static), which diagnoses failures across core agent abilities, and trajectory-level evaluation (HALO-Dynamic), where hallucinations arise within multi-step interactions and influence subsequent decisions. This progression reflects the increasing behavioral complexity of hallucination in multimodal agents.

Such trajectory-level hallucinations cannot be captured by HALO-Static, which evaluates each instance independently. Second, HALO-Dynamic studies trajectory-level hallucinations that arise during multi-step agent interactions. We formulate a step-level hallucination detection and classification task in which LVLMs must determine whether an agent's output at each step contains hallucination and identify its type based only on observable trajectory context.

We conduct extensive evaluations on both HALO-Static and HALO-Dynamic. For capability-level hallucination, we benchmark a wide range of open-source and proprietary LVLMs on HALO-Static, including QwenVL3.5 (Bai et al., 2025a), InternVL2.5 (Chen et al., 2024), and GPT-4o. To study trajectory-level hallucination, we collect agent trajectories from VisualWebArena (Koh et al., 2024) and manually annotate each step with hallucination presence and category labels, resulting in a dataset of 234 trajectories and 1,214 interaction steps. We then evaluate state-of-the-art multimodal models on this task and find that even strong LVLMs struggle to reliably detect hallucinations in agent trajectories.

Our contributions are summarized as follows: (1) We introduce HALO-Static, a capability-level benchmark for multimodal agent hallucination that systematically evaluates hallucination across core agent abilities, including perception, reasoning, memory, execution, communication, and multi-step decision making. (2) We formulate HALO-Dynamic, a step-level hallucination detection and classification task, where models must identify whether an agent's output is hallucinated and predict its type based only on observable trajectory context. (3) To support this task, we construct a trajectory-level dataset by collecting agent trajectories and manually annotating step-wise hallucination presence and categories. (4) We conduct extensive evaluations on both HALO-Static and HALO-Dynamic using a wide range of open-source and proprietary LVLMs, showing that cur-

rent models still exhibit significant hallucination across core agent capabilities and struggle to reliably detect hallucinations in multimodal agent trajectories.

2 Related Work

MLLMs as Agents Recent advancements have transformed Multimodal Large Language Models (MLLMs) from passive observers into active agents capable of autonomous decision-making and environmental interaction. This evolution is primarily driven by three core components: internal intelligence, tool invocation, and environment interaction (Yao et al., 2025). To enhance cognitive cores, internal intelligence methods such as Vision-R1 (Huang et al., 2025) and R1-VL (Zhang et al., 2025a) employ reinforcement learning (RL) to elicit long-chain reasoning during multimodal tasks. Complementary reflection mechanisms, either implicit (Meng et al., 2025) or explicit (Wan et al.), enable agents to verify and refine reasoning traces, while external memory systems (Yan et al., 2025b) facilitate long-term consistency beyond the context window. Furthermore, MLLMs extend their capabilities by proactively invoking external tools for information retrieval (Wu et al., 2025), complex mathematical coding (Gou et al., 2024), and active visual processingsuch as adaptive cropping and zooming to "think with images" (Zheng et al., 2025). These capabilities enable agents to engage with diverse surroundings, ranging from virtual GUI navigation (Luo et al., 2025) to physical embodied planning and manipulation (Yuan et al., 2025). However, these multi-step agentic trajectories are inherently prone to hallucinations.

Hallucinations in MLLMs Multimodal large language models (MLLMs) have achieved remarkable progress across a wide range of multimodal benchmarks (Liu et al., 2023; Bai et al., 2025b; Chen et al., 2024). Despite these advances, hallucination remains a persistent challenge. In MLLMs, hallucination typically refers to the generation of content that is inconsistent with, unsupported by, or unrelated to the visual input (Yan et al., 2025a; Jing et al., 2025a,b; Zhang et al., 2024). To quantify this issue, prior work has proposed a variety of evaluation metrics that measure the degree of image-text inconsistency, including FaithScore (Jing et al., 2024), CHAIR (Rohrbach et al., 2018), POPE (Li et al., 2023), and NOPE (Lovenia et al., 2023). Beyond evaluation, a growing body of research has explored mitigation strategies. These efforts span multiple directions, including decoding-time interventions (Leng et al., 2023), post-hoc correction mechanisms (Chang et al.), and curated high-quality training data to improve grounding fidelity. More recently, reinforcement learning-based approaches (Jing and Du) have been introduced to explicitly align visual and textual modalities. For instance, LLaVA-RLHF (Sun et al., 2023) leverages human preference signals to reduce hallucinated outputs through reward-driven optimization. While prior work studies hallucination in static vision-language tasks, multimodal agents operate in sequential, tool-augmented environments where hallucinations can propagate through reasoning, actions, and state transitions, making existing static LVLM evaluations insufficient.

3 HALO-Static: Capability-Level Hallucination Detection

We evaluate hallucination across six major capability areas and 12 fine-grained sub-types, corresponding to key components of multimodal agent behavior (Figure 2). Examples of QA instances for different hallucination categories are provided in Appendix K. For each sub-type below, we provide a definition of hallucination as a prediction that contradicts the verifiable evidence in the multimodal input or introduces unsupported information. All forms of hallucination follow a unified definition: a model prediction y is considered hallucinated if it contradicts or is unsupported by the verifiable evidence set E extracted from the multimodal input:

$$\text{Hall}(y) = 1 \quad \text{iff} \quad y \notin \Gamma(E),$$

where $\Gamma(E)$ denotes the space of facts entailed by the evidence. Below, we describe each category in natural language, focusing on the specific capability it stresses and typical error patterns it reveals.

Perception. Errors arising from the incorrect interpretation of visual content. 1) **OCR**: OCR hallucination typically manifests as the generation of spurious textual content or the misinterpretation of visible text regions within an image. To systematically evaluate this phenomenon, we construct a diagnostic test set derived from VisualWebBench (Liu et al.), leveraging its detailed annotations, including HTML structures and ground-truth OCR

transcriptions. Inspired by data augmentation techniques proposed in EDA (Wei and Zou, 2019), we generate three negative options for each instance using synonym replacement, random insertion, random swap, and random deletion operations. 2) **Element Grounding:** Element grounding hallucination occurs when a model refers to a UI element, bounding box, or spatial location that does not correspond to any object present in the image. To evaluate this behavior, we directly leverage VisualWebBench (Liu et al.), which provides annotated UI elements and their corresponding bounding boxes. For each instance, we construct negative options by sampling bounding boxes associated with other elements in the same image. 3) **General Image:** General image hallucination occurs when a model invents or misidentifies objects, attributes, or relations in real-world visual scenes (e.g., predicting "a cat on the table" when no such object exists). We directly adopt the POPE dataset (Li et al., 2023), which provides binary questions for evaluating object existence in images.

Execution. Errors involving environment interaction. 1) **Action:** Action hallucination occurs when the model selects an action that is inconsistent with the user intent or task goal, even if the action itself is valid in the current environment state. To evaluate this behavior, we construct instances based on SeeClick (Cheng et al., 2024), which provides GUI screenshots along with annotated UI elements and their associated functionalities for action prediction. We design multiple-choice questions based on these functionalities, using the correct element-action pair as the ground truth and sampling other elements from the same GUI page as negative options. 2) **API Calling:** API hallucination occurs when the model selects an incorrect tool or issues unsupported arguments, resulting in an API call that does not correspond to the intended execution given the current state and goal. We build this task based on the Text2Vis dataset (Rahman et al., 2025), which provides images along with Python code used to generate the corresponding visualizations. We use GPT-4o (prompts used can be found in Appendix H) to inject errors into the original code, such as incorrect arguments or syntactic mistakes, to generate three negative options.

Reasoning. Errors in task understanding, goal interpretation, or multi-step decision making. 1) **Goal Understanding:** Goal-understanding hallucination occurs when the model misinterprets the user's intended task or fails to respect explicit constraints specified in the instruction (e.g., solving a different problem or ignoring required conditions). To evaluate this behavior, we construct instances based on MIA-Bench (Qian et al., 2024), which provides complex instructions and evaluation criteria for multimodal agents. For each instruction, we prompt multiple strong MLLMs (Qwen2.5-VL, GPT-4o, Qwen3-VL, and InternVL) to generate candidate responses. Models are then asked to determine which response best satisfies the instruction's goal and constraints (prompts used can be found in Appendix H). 2) **Plan Generation:** Plan-generation hallucination occurs when the model produces a task plan that does not faithfully reflect the sequence of steps required to achieve the user's goal. This may include generating unnecessary or irrelevant actions, omitting required steps, or producing an order of operations that is logically invalid given the environment state. We construct this task based on TaPA (Wu et al., 2023), which provides planning tasks for embodied agents. To generate distractor options, we perturb the ground-truth plans by randomly deleting, repeating, or permuting steps to create three negative plans.

Memorization. Errors arising from the incorrect retrieval of external or internal knowledge. **1) External Retrieval:** External-retrieval hallucination arises when external lookup modules (e.g., long-video retrieval systems or document search tools) return incorrect, noisy, or irrelevant results, yet the model proceeds to answer confidently as if the retrieved content were correct. The hallucination thus stems from over-reliance on faulty external retrieval signals. We construct this task based on LVBench (Wang et al., 2025b), which provides long videos, multiple-choice questions, ground-truth answers, and corresponding temporal annotations. For evaluation, we extract a candidate video segment that may or may not contain the information required to answer the question. We further include an option indicating that the relevant information is not present in the retrieved segment, allowing us to test whether the model can correctly determine whether the retrieved content is relevant. **2) Internal Retrieval:** Internal-retrieval hallucination occurs when the agent selects an incorrect or irrelevant memory entry or video segment and subsequently bases its answer on this mismatched in-

ternal evidence, leading to grounded but incorrect responses. We also build this task using LVBench (Wang et al., 2025b). Given a long video and a question, the model must identify which temporal segment contains the relevant evidence. Candidate options correspond to different timestamps within the video, requiring the model to retrieve the correct segment before answering.

Communication. Errors arising from information exchange between agents, where an agent incorporates signals provided by other agents. 1) **Collaborative Perception:** Collaborative-perception hallucination occurs when another agent provides an incorrect perceptual description (e.g., misidentified objects or attributes), and the current agent trusts or propagates this erroneous perception, leading to downstream hallucinated conclusions. To evaluate this phenomenon, we construct instances based on MSCOCO (Lin et al., 2015), which provides paired images and captions. We synthesize incorrect perceptual statements using templates that modify object attributes or identities, and present outputs from multiple agents as candidate descriptions. The evaluated model must determine which agent's perception is consistent with the visual evidence. 2) **Agent-Bias Propagation:** Agent-bias propagation occurs when another agent introduces responses shaped by language priors or commonsense biases, and the current agent adopts or amplifies these biased signals instead of grounding its reasoning in the available visual evidence. We construct this task using VLind-Bench (Lee et al., 2025), which provides counterfactual imagetext pairs designed to expose multimodal biases. The model is given biased textual statements along with the corresponding images and must determine whether the textual claim is supported by the visual evidence.

Multi-Turn. All previous categories evaluate hallucinations in a single-turn QA setting, where each question is presented in isolation. In realistic agent deployments, however, hallucinations often emerge during multi-step interactions, where decisions depend on prior instructions, actions, and observations. To capture this phenomenon while maintaining a static QA evaluation format, we adopt a snapshot-style design inspired by MIRAGE-Bench (Zhang et al., 2025b). Specifically, we roll out agents on long-horizon tasks and identify decision steps where the context already

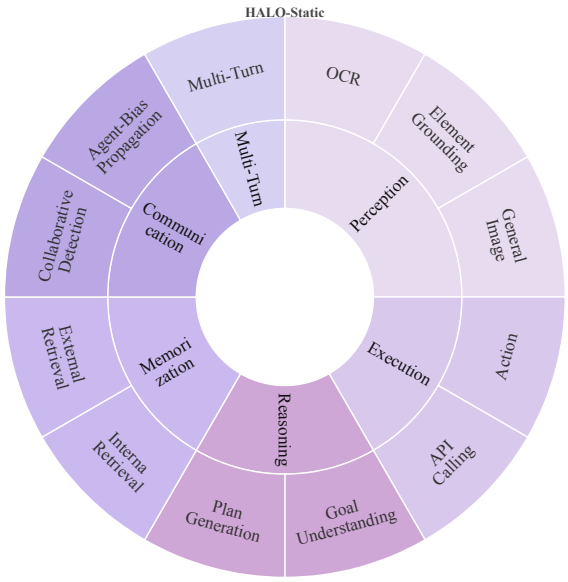


Figure 2: Sunburst visualization of the HALO-Static hallucination taxonomy, covering six capability areas and twelve fine-grained hallucination types.

contains sufficient evidence for a correct action but the agent still produces a hallucinated response. We then freeze such steps into self-contained QA instances. We construct this task based on trajectories generated by a multimodal agent (Wang et al., 2025a). For each selected step, we preserve the preceding instruction, action history, and observations as context, and formulate a multiple-choice question over the next decision. The correct option corresponds to the faithful action implied by the full interaction history, while distractor options are designed to reflect plausible but hallucinated decisions.

4 Analysis of HALO-Static

4.1 Models

To assess the capability-level hallucination in multimodal models, we employ a comprehensive set of baselines. **Open-source models** include LLaVA-1.5-7B, LLaVA-1.5-13B (Liu et al., 2023), LLaVA-1.6-7B, LLaVA-1.6-13B (Liu et al., 2024), InternVL2.5-1B, InternVL2.5-2B, InternVL3-4B, InternVL3-8B, InternVL2.5-38B, InternVL3-1B, InternVL3-2B (Chen et al., 2024), InternVL3-4B, InternVL3-8B, InternVL3-14B, InternVL3-38B (Zhu et al., 2025), DeepSeek-VL2-tiny, DeepSeek-VL2-small, DeepSeek-VL2 (Wu et al., 2024), Qwen3-VL-2B, Qwen3-VL-4B, Qwen3-VL-8B (Bai et al., 2025a), Qwen2.5-VL-3B, Qwen2.5-VL-7B, Qwen2.5-VL-32B and Qwen2.5-VL-72B

Model	Perceptron			Execution		Reasoning		Memorization		Communication		Multi-Turn	Average
	OCR	Ele.	GenI	Action	API.	Goal.	Plan.	Ext.	Int.	Col.	Bias.		
LLaVA-1.5-7B	58.42	10.65	41.00	6.80	17.33	11.25	23.33	–	–	86.72	7.33	68.33	33.12
LLaVA-1.5-13B	80.41	14.77	91.33	4.85	31.00	17.50	10.00	–	–	100.00	13.67	84.33	44.79
LLaVA-1.6-7B	63.23	11.38	89.00	7.77	17.33	7.50	0.00	–	–	80.91	9.33	62.33	34.88
LLaVA-1.6-13B	45.36	18.16	89.33	2.91	22.00	18.75	5.83	–	–	100.00	9.33	75.67	38.73
Qwen2.5-VL-3B	89.35	49.64	94.44	34.95	38.33	21.25	21.67	37.33	34.33	26.97	60.33	87.00	49.63
Qwen2.5-VL-7B	96.91	37.29	95.11	46.60	92.33	21.25	44.17	60.00	19.00	1.66	42.67	79.33	53.03
Qwen2.5-VL-32B	92.44	24.46	91.11	2.91	97.33	18.75	57.50	35.33	18.67	83.40	18.67	30.33	47.57
Qwen2.5-VL-72B	98.63	55.45	94.33	16.50	99.67	32.50	75.00	57.67	29.33	97.10	44.33	89.23	65.81
Qwen3-VL-2B	90.03	26.15	95.22	58.25	82.33	32.50	47.50	44.33	28.33	29.88	41.00	70.67	53.85
Qwen3-VL-4B	93.13	60.05	94.56	83.50	95.67	18.75	55.83	41.33	17.00	39.00	43.67	78.33	60.07
Qwen3-VL-8B	78.01	54.72	95.00	38.83	99.33	33.75	65.00	40.00	25.67	32.37	55.00	60.00	56.47
InternVL2.5-1B	59.45	12.11	83.00	4.85	20.00	8.75	15.83	29.33	22.00	32.37	23.00	69.33	31.67
InternVL2.5-2B	77.32	15.98	90.78	1.94	29.00	15.00	25.00	45.00	22.00	5.81	7.00	57.00	32.65
InternVL2.5-4B	81.10	15.98	88.11	5.83	86.67	15.00	33.33	32.67	24.67	29.05	21.67	77.00	42.59
InternVL2.5-8B	86.25	31.96	70.56	10.68	61.33	26.25	27.50	30.67	29.33	35.27	52.00	75.33	44.76
InternVL2.5-38B	86.94	47.46	90.56	7.77	97.33	36.25	48.33	32.00	30.67	88.38	43.67	80.33	57.47
InternVL3.5-1B	68.73	21.07	81.11	0.00	44.67	30.00	22.50	30.67	35.00	17.01	17.67	51.89	35.03
InternVL3.5-2B	76.29	32.69	88.89	19.42	43.67	15.00	42.50	32.00	18.67	55.19	47.00	19.33	40.89
InternVL3.5-4B	95.19	72.64	88.22	7.77	81.67	21.25	29.17	31.33	26.67	62.24	71.00	50.00	53.10
InternVL3.5-8B	90.38	72.64	87.00	10.68	89.33	13.75	39.17	37.00	30.33	56.43	73.33	44.33	53.70
InternVL3.5-14B	95.19	87.17	91.33	0.97	92.67	36.25	50.83	36.67	28.00	86.72	51.00	56.33	59.43
InternVL3.5-38B	96.22	86.92	94.44	2.91	97.00	30.00	60.83	39.00	25.67	97.10	71.33	55.33	63.06
DeepSeek-VL2-small	33.68	11.14	73.89	54.37	5.33	10.00	25.00	–	–	23.24	18.67	61.00	31.63
DeepSeek-VL2-tiny	71.13	14.53	93.44	24.27	15.33	13.75	21.67	–	–	84.65	0.67	51.67	39.11
DeepSeek-VL2	52.92	11.86	93.89	46.60	13.85	5.00	15.83	–	–	24.48	20.33	52.33	33.71
GPT-4o	96.56	31.72	89.78	90.29	94.67	28.75	51.67	39.39	33.33	93.36	48.00	86.33	57.13
GPT-5.1	99.66	65.38	91.33	77.67	99.67	45.00	56.67	60.61	37.67	98.76	52.67	84.67	72.48

Table 1: The performance comparison between different baselines across all hallucinations of our HALO-Static. All numbers denote accuracy (%). – denotes the model cannot be evaluated on the corresponding task due to lack of video input support.

(Bai et al., 2025b). For **closed-source models**, we evaluate against GPT-4o and GPT-5.1. Detailed model configurations.

4.2 Results

Results Overview. Table 1 summarizes HALO-Static accuracy across 12 fine-grained hallucination sub-types. Based on the observation on open-source models, stronger models achieve higher average performance, with **Qwen2.5-VL-72B** reaching the best average (**65.81**), followed by **InternVL3.5-38B** (63.06) and **Qwen3-VL-4B** (60.07). While scaling generally helps, gains are not uniform across capabilities: e.g., within Qwen2.5, the 32B model (47.57) trails the 7B model (53.03), suggesting that agent reliability depends not only on capacity but also on capability-specific training and alignment. For closed-source models, **GPT-5.1** achieves the strongest overall performance (**72.48**), substantially outperforming all open-source counterparts, while **GPT-4o** obtains competitive results (57.13) comparable to mid-to-large open-source models. Despite their overall strength, even these advanced proprietary models still exhibit noticeable weaknesses in several hallucination categories such as reasoning and memorization, indicating that hallucination in agent-style multimodal settings remains an open challenge.

Perception vs. Grounding. Perception-heavy sub-tasks (OCR and general image understanding) are relatively strong for most modern MLLMs (e.g., Qwen2.5-VL-72B achieves 98.63 on OCR and 94.33 on GenI; InternVL3.5-38B achieves 96.22 on OCR and 94.44 on GenI). In contrast, element grounding remains substantially harder and varies widely across model families (e.g., Qwen2.5-VL-32B: 24.46 vs. InternVL3.5-14B/38B: 87.17/86.92), indicating that hallucinations in UI/spatial referencing cannot be reduced to OCR-style recognition errors.

Execution is the Primary Bottleneck. Execution errors are the most persistent failure mode: the **Action** column is extremely low for many otherwise capable models (e.g., Qwen2.5-VL-32B: 2.91; InternVL3.5-14B: 0.97; InternVL3.5-38B: 2.91), revealing frequent hallucinations as *invalid or infeasible actions* under the current environ-

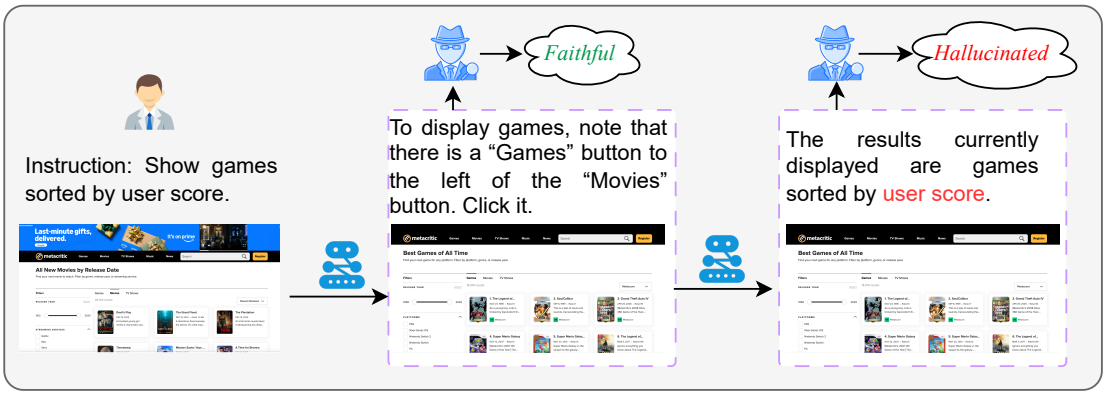


Figure 3: Illustration of the step-level hallucination detection task, where each intermediate step is evaluated for faithfulness to the instruction, visual interface, and observed page state.

ment state. Meanwhile, **API Calling** is often high for top models (e.g., Qwen2.5-VL-72B: 99.67; Qwen2.5-VL-32B: 97.33; InternVL3.5-38B: 97.00), suggesting that selecting the correct action or tool is more challenging than producing syntactically valid interaction steps, indicating that the primary difficulty lies in high-level decision making rather than low-level action formatting. More analysis can be found in Appendix A.

5 HALO-Dynamic: Trajectory-Level Hallucination Detection

Although the HALO-Static benchmark provides a comprehensive view of hallucination across perception, reasoning, memory, execution, communication, and multi-turn understanding, it evaluates models in an *isolated* setting. This captures a model's *capabilities*, but real multimodal agents operate through *long trajectories*: they update their beliefs, observe changing environments, interact with tools, and generate action sequences. Therefore, we move from capability assessment to *trajectory-level reliability*, introducing HALO-Dynamic: a step-level hallucination detection task that evaluates whether an agent hallucinates at each point **during** its interaction with the environment. HALO-Dynamic addresses a complementary question: *Given an agent's multimodal observations, history, and current output, can we detect whether the agent is hallucinating?* This dynamic formulation aligns with real deployment scenarios, where systems must be monitored online under black-box constraints.

5.1 Problem Formulation

We consider a multimodal agent interacting with an environment over a trajectory of length T. At each step t, the agent receives a multimodal observation O_t (e.g., an image, a webpage screenshot, or the result of a tool invocation), and a history of all previous interactions: $H_t = \{(O_1, Y_1), (O_2, Y_2), \ldots, (O_{t-1}, Y_{t-1})\}$. The agent then produces an output $Y_t = (R_t, A_t)$, where R_t is a natural language response and A_t is an action (e.g., clicking an element).

A step-level hallucination occurs when the agent's current output Y_t either invents unsupported facts that cannot be grounded in the available evidence $\{H_t, O_t\}$, or contradicts verifiable state information encoded in the environment metadata or oracle trajectories. The step-level hallucination detection task is to construct a detector $D(H_t, O_t, Y_t) \rightarrow \{0, 1\}$, which predicts whether the agent's output at step t contains hallucination (Figure 3) as follows:

$$h_t = \begin{cases} 1, & \text{if } Y_t \text{ contains hallucination,} \\ 0, & \text{otherwise.} \end{cases}$$

Furthermore, if a step is detected as hallucinated, the detector is further required to predict the hallucination category according to the taxonomy introduced in Section 3.

5.2 Benchmark Construction

We construct our benchmark by sampling 234 GPT-4o trajectories from VisualWebArena (Koh et al., 2024), comprising **1,214** interaction steps in total. We recruit **three** annotators to label the data at the step level, and ensure that **each step is**

Model	Hallucination	Category
LLaVA-1.5-7B	35.8%	1.8%
LLaVA-1.5-13B	35.7%	4.9%
LLaVA-1.6-7B	35.4%	0.0%
LLaVA-1.6-13B	35.5%	3.4%
Qwen2.5-VL-3B	35.7%	5.5%
Qwen2.5-VL-7B	35.3%	0.0%
Qwen2.5-VL-32B	46.5%	19.0%
Qwen2.5-VL-72B	53.7%	23.4%
Qwen3-VL-2B	56.6%	22.2%
Qwen3-VL-4B	40.3%	4.9%
Qwen3-VL-8B	52.7%	30.4%
InternVL2.5-1B	43.0%	1.8%
InternVL2.5-2B	48.6%	7.2%
InternVL2.5-4B	47.5%	7.2%
InternVL2.5-8B	49.1%	3.2%
InternVL2.5-38B	57.8%	15.7%
InternVL3.5-1B	47.9%	8.7%
InternVL3.5-2B	45.4%	**53.5%**
InternVL3.5-4B	51.6%	28.1%
InternVL3.5-8B	49.0%	29.0%
InternVL3.5-14B	**63.4%**	25.5%
InternVL3.5-38B	54.2%	22.0%
DeepSeek-VL2-small	35.4%	0.0%
DeepSeek-VL2-tiny	35.4%	0.0%
DeepSeek-VL2	35.4%	0.0%
GPT-4o	61.3%	36.7%
GPT-5.1	56.8%	24.1%

Table 2: Step-level hallucination detection on agent trajectories (HALO-Dynamic). We compare models for detecting hallucinations at each trajectory step.

annotated by all three annotators. Before the full-scale annotation, we first run a small qualification test on a pilot set of **10** trajectories (**37** steps); only annotators who meet our quality requirement on this test set are hired for the main annotation. For every step, we provide annotators with the *trajectory history* and the *current step* context, along with the definition of each category in our taxonomy. Annotators are instructed to (i) determine whether the step contains hallucination, (ii) if yes, assign the corresponding hallucination category, and (iii) provide a brief explanation to justify their decision. We aggregate the three annotations using majority voting to obtain the final label for each step (both the hallucination flag and, when applicable, the category).

6 Analysis of HALO-Dynamic

To assess the performance of MLLMs as judges for trajectory-level hallucination detection, we employ the same baselines evaluated in Section 4.

Table 2 presents step-level hallucination detection results on HALO-Dynamic. Overall, larger and more capable models tend to achieve higher hallucination detection accuracy. Among open-source models, InternVL3.5-14B achieves the best detection performance (63.4%), followed by InternVL2.5-38B (57.8%) and Qwen3-VL-2B (56.6%). However, predicting hallucination categories is substantially more challenging: most open-source models obtain below 30% accuracy. Notably, InternVL3.5-2B achieves the highest category accuracy (53.5%), suggesting that smaller models may occasionally exhibit stronger calibration for fine-grained classification despite weaker detection accuracy.

Closed-source models remain competitive, with GPT-4o achieving strong performance on both hallucination detection (61.3%) and category prediction (36.7%). In contrast, DeepSeek-VL2 variants consistently obtain near-random detection performance (35.4%) with zero category accuracy, indicating difficulty in following the hallucination detection task format. Overall, the results suggest that identifying hallucination types in dynamic agent trajectories remains challenging for current multimodal models.

7 Conclusion

In this work, we study hallucination in multimodal agents and introduce HALO, a benchmark designed to evaluate hallucination beyond static vision-centric settings. HALO consists of two complementary components: HALO-Static, a capability-level benchmark that measures hallucination across core agent abilities such as perception, reasoning, memory, execution, and communication, and HALO-Dynamic, a step-level hallucination detection and classification task defined over agent interaction trajectories. To support trajectory-level evaluation, we construct a manually annotated dataset of agent trajectories collected from VisualWebArena with step-wise hallucination labels. Extensive experiments on a wide range of open-source and proprietary LVLMs reveal that current models still exhibit substantial hallucination across core agent capabilities and struggle to reliably detect hallucinations in agent trajectories. We hope HALO provides a useful benchmark for studying hallucination in multimodal agents and encourages future research on improving the reliability and robustness of agentic multimodal systems.

Limitations

HALO-Dynamic is constructed from agent trajectories collected in VisualWebArena, which provides realistic multimodal web-based interaction scenarios. While this setting captures complex agent behaviors such as perception, reasoning, and action execution, it represents only one type of agent environment. Future work could extend HALO-Dynamic to additional agent platforms and environments to further study hallucination behaviors in broader multimodal agent settings.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. 2025a. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025b. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Yue Chang, Liqiang Jing, Xiaopeng Zhang, and Yue Zhang. A unified hallucination mitigation framework for large vision-language models. *Transactions on Machine Learning Research*.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. 2024. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

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.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. 2024. Tora: A tool-integrated reasoning agent for mathematical problem solving.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. 2025. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*.

Liqiang Jing, Guiming Hardy Chen, Ehsan Aghazadeh, Xin Eric Wang, and Xinya Du. 2025a. A comprehensive analysis for visual object hallucination in large vision-language models. *arXiv preprint arXiv:2505.01958*.

Liqiang Jing and Xinya Du. Fgaif: Aligning large vision-language models with fine-grained ai feedback. *Transactions on Machine Learning Research*.

Liqiang Jing, Viet Lai, Seunghyun Yoon, Trung Bui, and Xinya Du. 2025b. Fifa: Unified faithfulness evaluation framework for text-to-video and video-to-text generation. *arXiv preprint arXiv:2507.06523*.

Liqiang Jing, Ruosen Li, Yunmo Chen, and Xinya Du. 2024. Faithscore: Fine-grained evaluations of hallucinations in large vision-language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 5042–5063. Association for Computational Linguistics.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. 2024. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 881–905. Association for Computational Linguistics.

Kang-il Lee, Minbeom Kim, Seunghyun Yoon, Minsung Kim, Dongryeol Lee, Hyukhun Koh, and Kyomin Jung. 2025. Vlind-bench: Measuring language priors in large vision-language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4129–4144.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2023. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. *CoRR*, abs/2311.16922.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 292–305.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft coco: Common objects in context.

Xixun Lin, Yucheng Ning, Jingwen Zhang, Yan Dong, Yilong Liu, Yongxuan Wu, Xiaohua Qi, Nan Sun, Yanmin Shang, Kun Wang, Pengfei Cao, Qingyue Wang, Lixin Zou, Xu Chen, Chuan Zhou, Jia Wu, Peng Zhang, Qingsong Wen, Shirui Pan, Bin Wang,

Yanan Cao, Kai Chen, Songlin Hu, and Li Guo. 2025. Llm-based agents suffer from hallucinations: A survey of taxonomy, methods, and directions.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *CoRR*, abs/2304.08485.

Junpeng Liu, Yifan Song, Bill Yuchen Lin, Wai Lam, Graham Neubig, Yuanzhi Li, and Xiang Yue. Visualwebbench: How far have multimodal llms evolved in web page understanding and grounding? In *First Conference on Language Modeling*.

Holy Lovenia, Wenliang Dai, Samuel Cahyawijaya, Ziwei Ji, and Pascale Fung. 2023. Negative object presence evaluation (nope) to measure object hallucination in vision-language models.

Run Luo, Lu Wang, Wanwei He, Longze Chen, Jiaming Li, and Xiaobo Xia. 2025. Gui-r1 : A generalist r1-style vision-language action model for gui agents.

Muhammad Maaz, Hanoona Abdul Rasheed, Salman Khan, and Fahad Khan. 2024. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 12585–12602. Association for Computational Linguistics.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, et al. 2025. Mm-eureka: Exploring visual aha moment with rule-based large-scale reinforcement learning. *arXiv preprint arXiv:2503.07365*.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2024. Gaia: a benchmark for general ai assistants. In *International Conference on Learning Representations*, volume 2024, pages 9025–9049.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. 2018. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8494–8502.

Yusu Qian, Hanrong Ye, Jean-Philippe Fauconnier, Peter Grasch, Yinfei Yang, and Zhe Gan. 2024. Mia-bench: Towards better instruction following evaluation of multimodal llms. *arXiv preprint arXiv:2407.01509*.

Mizanur Rahman, Md Tahmid Rahman Laskar, Shafiq Joty, and Enamul Hoque. 2025. Text2vis: A challenging and diverse benchmark for generating multimodal visualizations from text. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 31837–31862.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045. Association for Computational Linguistics.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. 2023. Aligning large multimodal models with factually augmented RLHF. *CoRR*, abs/2309.14525.

Zhongwei Wan, Zhihao Dou, Che Liu, Yu Zhang, Dongfei Cui, Qinjian Zhao, Hui Shen, Jing Xiong, Yi Xin, Yifan Jiang, et al. Srpo: Enhancing multimodal llm reasoning via reflection-aware reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Haozhe Wang, Alex Su, Weiming Ren, Fangzhen Lin, and Wenhu Chen. 2025a. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. *arXiv preprint arXiv:2505.15966*.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. 2025b. Lvbench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22958–22967.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 6382–6388.

Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. 2025. Mmsearch-r1: Incentivizing lmms to search. *arXiv preprint arXiv:2506.20670*.

Zhenyu Wu, Ziwei Wang, Xiuwei Xu, Jiwen Lu, and Haibin Yan. 2023. Embodied task planning with large language models. *arXiv preprint arXiv:2307.01848*.

Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, et al. 2024. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*.

Bowen Yan, Zhengsong Zhang, Liqiang Jing, Eftekhar Hossain, and Xinya Du. 2025a. Fiha: Automated fine-grained hallucinations evaluations in large vision language models with davidson scene graphs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12014–12026.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Kristian Kersting, Jeff Z Pan, Hinrich Schütze, et al. 2025b. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *arXiv preprint arXiv:2508.19828*.

Huanjin Yao, Ruifei Zhang, Jiaxing Huang, Jingyi Zhang, Yibo Wang, Bo Fang, Ruolin Zhu, Yongcheng Jing, Shunyu Liu, Guanbin Li, et al. 2025. A survey on agentic multimodal large language models. *arXiv preprint arXiv:2510.10991*.

Yifu Yuan, Haiqin Cui, Yaoting Huang, Yibin Chen, Fei Ni, Zibin Dong, Pengyi Li, Yan Zheng, and Jianye Hao. 2025. Embodied-r1: Reinforced embodied reasoning for general robotic manipulation. *arXiv preprint arXiv:2508.13998*.

Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. 2025a. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization.

Weichen Zhang, Yiyou Sun, Pohao Huang, Jiayue Pu, Heyue Lin, and Dawn Song. 2025b. Mirage-bench: Llm agent is hallucinating and where to find them. *arXiv preprint arXiv:2507.21017*.

Yue Zhang, Jingxuan Zuo, Ke Su, and Liqiang Jing. 2024. Fine-grained and explainable factuality evaluation for multimodal summarization. *arXiv preprint arXiv:2402.11414*.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. 2025. Deepeyes: Incentivizing "thinking with images" via reinforcement learning.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models.

A More Experimental Analysis

Reasoning, Memory, and Multi-Turn Reliability. Reasoning and memorization remain only moderately reliable: plan generation improves with stronger models (e.g., Qwen2.5-VL-72B: 75.00; InternVL3.5-38B: 60.83; Qwen2.5-VL-32B: 57.50) but is far from solved, and internal retrieval is consistently low (e.g., Qwen2.5-VL-32B: 18.67; Qwen2.5-VL-72B: 29.33; InternVL3.5-38B: 25.67), suggesting frequent failures in selecting correct history/memory evidence. Multi-turn consistency is comparatively strong for larger models (e.g., Qwen2.5-VL-72B: 89.23; InternVL2.5-38B: 80.33) yet still imperfect, supporting HALOs focus on multi-stage, state-dependent hallucinations that can emerge and propagate across decision steps.

B Quality Control of HALO-Static

To ensure HALO-Static measures hallucination rather than generic multiple-choice (MC) robustness, we implemented a rigorous human-in-the-loop verification pipeline during data construction.

- *Manual Filtering: For the 3,517 instances, human annotators manually screened the automatically generated distractors to ensure they are: (1) linguistically plausible, and (2) strictly contradicted by the verifiable evidence E.*
- *Artifact Elimination: We specifically removed instances where the correct answer could be inferred solely through internal linguistic logic or common sense without grounding in the multimodal input.*
- *Human Baseline: Preliminary human trials from 200 random samples show a near-perfect accuracy (>95%) when provided with evidence, contrasting sharply with the significantly lower performance of current state-of-the-art LVLMs (e.g., GPT-4o at 56.75%). This performance gap demonstrates that the failures observed are genuine hallucinations rooted in capability deficits, not dataset-specific artifacts.*

For both HALO-Static and HALO-Dynamic, we performed manual verification: invalid, ambiguous, or low-quality generated samples were removed, and only samples consistent with the intended hallucination categories and task definitions were retained.

C Taxonomy and Operational Boundaries

To ensure conceptual rigor and minimize ambiguity between the core capability areas in HALO-Static, we provide formal operational definitions according to (Lin et al., 2025). We implement a Primary Root-Cause Criterion for both data construction and human annotation: a hallucination is assigned to the earliest stage in the interaction pipeline where the failure originates.

- *Reasoning hallucinations refer to the phenomenons where LLM-based agents generate seemingly plausible plans which are in fact logically flawed or unsupported.*
- *Execution hallucinations refer to the phenomenon where LLM-based agents claim to have completed certain sub-stages during the execution phase, but in reality, they have not actually been performed or accomplished.*
- *Perception hallucinations refer to the phenomenon where, during the process of receiving and transforming external information, an agent produces internal observations that significantly deviate from or contain factual errors about the actual learning environment. This typically arises from environmental sensor malfunction or limited encoding capability. Such hallucinations undermine the agents accurate perception and further disrupt subsequent decision-making processes*
- *Memorization hallucinations refer to the phenomenon in which an agent implicitly assumes its memory to be accurate and reliable, without validating the correctness of the stored content. Consequently, when the agent incorporates its memory module into decisionmaking, it may rely on outdated, fabricated, or conflated memories, resulting in flawed reasoning or inappropriate actions.*
- *Communication hallucinations refer to the phenomenon in which LLM-based agents appear to engage in meaningful inter-agent communication, yet the information exchanged is inaccurate, misleading, or fabricated, thereby undermining collaboration.*

D More HALO-Dynamic Analysis and Annotation Reliability

D.1 Annotation Quality and Reliability

To ensure the high fidelity of the HALO-Dynamic benchmark, we recruited three professional annotators to evaluate each trajectory step independently. All annotators were required to pass a qualification test on a pilot set before the full-scale annotation. Final labels were determined via a majority voting mechanism for both hallucination presence and category classification.
To quantify inter-annotator agreement, we computed Fleiss κ across all annotators, obtaining a score of 0.7597. This score indicates "substantial agreement" and supports the reliability and consistency of our step-level annotations. For every step identified as a hallucination, annotators also provided a brief explanation to justify their decision.

D.2 Dataset Statistics and Distribution

HALO-Dynamic comprises 234 trajectories collected from VisualWebArena, totaling 1,214 interaction steps. The average trajectory length is 5.19 steps, with a median of 3 and a maximum of 31 steps.
The dataset presents a representative distribution of agentic failures in multimodal environments:

- ***Hallucination Presence**: At the step level, 64.58% are labeled as hallucination and 35.42% as non-hallucination.*
- ***Category Distribution**: Trajectory-level hallucinations are primarily driven by Reasoning errors (89.41%), followed by Execution (5.48%), Perception (4.46%), and Memorization (0.51%).*

This distribution highlights that hallucinations in interactive agent trajectories are predominantly rooted in high-level reasoning failures rather than isolated perceptual errors.

D.3 Multi-dimensional Performance Analysis

We further analyze the detection performance of state-of-the-art models across different trajectory lengths and categories to identify specific challenges in online monitoring.

D.3.1 History Length Analysis

As shown in Table 3, detection performance does not monotonically degrade with longer trajectories. While medium-length trajectories (36 steps) prove the most challenging (50.00% accuracy), accuracy improves for longer trajectories (63.50%), suggesting that hallucinations may become more explicit or accumulate over time as the environment state diverges from the agent's beliefs.

D.3.2 Category-level Performance

Detection accuracy varies significantly across hallucination types (Table 4). Reasoning achieves the highest detection accuracy (46.02%), while execution (27.27%) and perception (18.18%) remain significantly more challenging. This indicates that fine-grained perceptual inconsistencies and action-intent mismatches are harder for current models to detect reliably compared to high-level logical errors.

Table 3: Step-level detection accuracy across different history lengths.

History Length	# Samples	Accuracy (%)
1–2 Steps	437	56.98%
3–6 Steps	388	50.00%
7+ Steps	389	63.50%
Overall	**1214**	**56.84%**

Table 4: Category-level detection performance on HALO-Dynamic.

Hallucination Category	Accuracy (%)
Reasoning	46.02%
Execution	27.27%
Perception	18.18%
Memorization	0.00%

E Evaluation Protocols and Reproducibility

To provide a clear understanding of our benchmarking process and ensure reproducibility, we detail the evaluation metrics and experimental setups for both HALO-Static and HALO-Dynamic.

E.1 HALO-Static: Multiple-Choice Accuracy

For the HALO-Static benchmark, each of the 3,517 instances is formulated as a multiple-choice question-answering (QA) task. We evaluate model

*performance using **Accuracy**, defined as the percentage of correctly selected answers across all instances in a given sub-task:*

$$Accuracy_{Static} = \frac{\sum_{i=1}^{N} \mathbb{I}(\hat{y}_i = y_i)}{N} \times 100\% \quad (1)$$

where N is the total number of test cases, $\hat{y}_i$ is the model's predicted option, and y_i is the ground-truth label. This metric provides a straightforward assessment of a model's robustness against specific hallucination types at the capability level.

E.2 HALO-Dynamic: Step-Level Metrics

*HALO-Dynamic evaluates the reliability of multi-modal agents during real-time interactions across 1,214 steps. The evaluation is conducted at the **step level** using two primary metrics:*

1. ***Hallucination Detection Accuracy (Binary):** This measures the model's ability to identify the presence of a hallucination at each decision step ($h_t \in \{0, 1\}$). Given the class distribution (64.58% hallucinated steps), we additionally report majority-class baselines in our analysis to provide a balanced context for detector performance.*

2. ***Category Prediction Accuracy:** For steps identified as hallucinations, the model must further predict the correct category from our taxonomy. Accuracy is calculated based on the alignment between the predicted and ground-truth categories.*

E.3 Model Configurations and Data Availability

Following the reviewer's observation, we evaluated a comprehensive set of 27 models, comprising 25 open-source Large Vision-Language Models (LVLMs) and 2 proprietary models (GPT-4o and GPT-5.1). All models were tested using their official API or inference configurations to ensure fair comparison.

Upon acceptance, we will release the complete HALO benchmark, including all 3,517 HALO-Static QA pairs, the 234 annotated trajectories of HALO-Dynamic, and the full evaluation suite. The dataset download links and source code will be provided in the final version of the manuscript.

F Additional Model Evaluation

To further validate the generality of our findings, we additionally evaluate Claude 4.6-Sonnet [1] and Grok 4.20 [2] on HALO-Dynamic. Claude achieves 69.6% hallucination detection accuracy and 23.2% category accuracy, while Grok achieves 61.0% and 28.8%, respectively. These results show that closed-source models remain competitive, while still exhibiting limitations in fine-grained category prediction. This further supports that our conclusions are not specific to a particular model family.

G Cross-Domain Pilot Test: Generality to Diverse Multimodal Agent Environments

To demonstrate that the insights, taxonomy, and evaluation protocols of HALO-Dynamic generalize beyond web-based GUI interactions, we conducted a cross-domain pilot study. We collected and manually annotated an additional 40 trajectories (totaling 210 interaction steps) generated by the open-source `Qwen2.5-VL-72B` model across two distinct, authoritative multi-modal agent benchmarks representing entirely different environmental modalities: 1) Embodied Physical Planning (Puig et al., 2018) tasks require an agent to process 3D egocentric visual observations, maintain spatial reasoning of the physical surroundings, and generate sequential actionable commands (e.g., navigating rooms and manipulating household objects) under long-horizon constraints. 2) Multimodal Document Intelligence (Mialon et al., 2024) tasks present high-level reasoning challenges where the agent must parse high-resolution visual infographics, financial reports, or multi-page PDFs, and sequentially invoke data-analysis tools or Python code blocks to synthesize final answers.

***Cross-Domain Hallucination Distribution** As illustrated in Table 5, the natural distribution of hallucinations aggregated across these distinct benchmarks firmly preserves our core architectural insights. While the combined pilot data reflects highly diverse environmental modalities and action spaces, Reasoning hallucinations remain the absolute dominant failure mode across all domains (accounting for 76.19% in the pilot test com-*

[1] https://www.anthropic.com/news/claude-sonnet-4-6.
[2] https://docs.x.ai/developers/models/grok-4.20

pared to 89.41% in the main dataset). This consistent trend across different platforms and model architectures proves that cognitive-level logical breakdowns are a universal vulnerability in multi-step multimodal agents, strongly validating the platform-agnostic design of the HALO framework.

Online Detection Consistency Across Environments *We further deployed GPT-4o and Qwen2.5-VL-72B individually as online detectors on these new cross-domain steps. Specifically, GPT-4o achieved a step-level detection accuracy of 55.24%, while Qwen2.5-VL-72B achieved 52.62%, yielding a combined average accuracy of 53.93%. These individual performances highly align with the baselines achieved on our main Web-GUI dataset (57.50% overall).*

H Prompts

Hallucination Dectection

You are a strict evaluator for hallucinations in agent trajectories. Your job is to decide whether the CURRENT STEP contains hallucination, using only the evidence provided in the trajectory (previous steps + current step). Do not use outside knowledge. Do not assume UI elements exist unless they are explicitly shown/described.
Input you will receive
I will provide: - Task/Intent: the overall goal - History (previous steps): a sequence of earlier steps (UI state/screenshot description, observation, reasoning, action, output, and post-action new UI state if available) - Current step: the step to evaluate (same fields)
Key rule: You must judge the current step, not the entire trajectory.
Hallucination Categories (label the current step)
Mark the current step as hallucination if any of the following holds:
Perception Hallucination The agent claims to see/read UI content that is not present, or misreads a critical UI element (text/value/button/state).
Evidence: contradiction with the current UI/screenshot description or observation.
Execution Hallucination
The plan/intent is correct, but the action is wrong or malformed (wrong click/typing/selection, wrong field, failure to clear old text, wrong order of operations, etc.), causing the step not to implement what the reasoning claims.
Evidence: mismatch between reasoning/plan and the actual action; or post-action UI does not reflect the intended change when the new UI state/observation supports that.
Reasoning Hallucination
The agent draws an unsupported or logically invalid conclusion from the available UI evidence (e.g., claims the task is completed, identifies the most expensive item without comparing prices, asserts constraints are satisfied without evidence).
Evidence: the conclusion is not entailed by the provided UI/observation, or contradicts it.
Planning Hallucination
The agents plan/order of steps violates the task constraints or omits necessary constraints such that following the plan would likely fail (e.g., should sort by date before filtering but plans the opposite; ignores required color/price/location constraints).
Evidence: compare plan steps against Task/Intent constraints.
Memory Hallucination
The agent incorrectly references prior steps (e.g., thinks a filter was applied, login completed, item added, or page is X when history shows otherwise).
Evidence: conflict with the provided History.
Communication Hallucination (only if the step includes user-facing statements) The agent tells the user it did something or found a result that is not supported by the trajectory evidence.
Evidence: unsupported claim or contradiction with UI/history.
Evaluation requirements (very important)
Use only the provided fields. No guessing.
Output format (STRICT JSON ONLY)
Return exactly one JSON object and nothing else.
"hallucination": "yes or no",
"category": "none | perception | execution | reasoning | memorization | communica-

Table 5: Comparative hallucination category distribution between the main Web-GUI dataset and the combined Cross-Domain Pilot dataset.

	VisualWebArena		Cross-Domain Pilot	
Hallucination Category	#Steps	Percentage (%)	#Steps	Percentage (%)
Reasoning	1085	89.37%	160	76.19%
Execution	67	5.52%	32	15.24%
Perception	54	4.45%	15	7.14%
Memorization	8	0.66%	3	1.43%
Total Annotated Steps	1,214	100.00%	210	100.00%

tion",
"rationale": "One-sentence summary of why",

Begin evaluation now
Task:
{TASK INTENT}
History (previous steps): {HISTORY STEPS}
Current step: {CURRENT STEP}

Prompt for Negative Code Generation

Prompt: You are an assistant that creates negative (incorrect) matplotlib code options for testing code understanding and function-calling capabilities.
You are given a correct matplotlib code snippet (ground truth).
Generate exactly **5 negative code snippets** that are plausible variations but wrong.
The negative codes can include: 1. Semantic / Visual mistakes wrong variables, swapped x/y, wrong color map, axis flipped, etc. 2. Syntax mistakes invalid function names, missing commas or parentheses, wrong parameter names. 3. Logical mistakes wrong computation, wrong aggregation, missing labels, plotting wrong data.
Rules: - Make sure every snippet looks realistic not random gibberish. - Some can run but produce wrong plots, others can raise errors. - Do not repeat the ground truth exactly. - For each, briefly explain why its wrong.
Output exactly in this JSON format:

"ground truth": "<the input code>", "negative choices": ["code": "...", "why wrong": "...", "code": "...", "why wrong": "...", "code": "...", "why wrong": "...", "code": "...", "why wrong": "...", "code": "...", "why wrong": "..."]

Prompt for LLM-based Evaluator

Image: ${image}$

Prompt: Here is an instruction for a multimodal LLM: "<instruction>"
You need to grade if the response from the model follows each component of the instruction.
The components are: - Component 1: "<component 1>" - Component 2: "<component 2>" ...
Each component has a corresponding weight.
The response is: "<model response>"
You need to score the response and be strict.
The total score ranges from 0 to 10.
Output in the following format: score of component 1: x/?, score of component 2: y/?, ..., total score: z/10.
Then explain your reasoning.

I Usage of LLMs

We used a large language model (LLM) solely for minor language editing and polishing to improve the clarity and readability of the manuscript. The LLM was not used for generating research ideas, conducting experiments, analyzing results, or writing technical content. All research design, experiments, and analyses were conducted by the authors.

J Data Construction

For the newly constructed data, we conduct human verification to remove low-quality instances. The data statistics of HALO-Static are shown in Table 6.

K Quatative Examples

We show one representative example for each task in Figure 4 and Figure 5.

Type	Subtype	#Cases
Perception	OCR	291
	Element Grounding	413
	General Image	900
Execution	Action	103
	API Calling	410
Reasoning	Goal Understanding	80
	Plan Generation	120
Memorization	External Retrieval	300
	Internal Retrieval	300
Communication	Collaborative Perception	300
	Agent-Bias Propagation	300
Multi-Turn	Multi-Turn	300
Total		**3,517**

Table 6: Distribution of cases across fine-grained hallucination types in HALO-Static.

L Guidelines for Annotators

Multimodal Agent–GUI Trajectory Annotation Instructions (Google Sheet)

1) Purpose

You will annotate **hallucinations** in multimodal agent–GUI interaction trajectories.
Each trajectory is a sequence of steps where an agent observes a GUI, reasons, and takes actions.

Your task for each step (each row):

1. Decide whether a hallucination occurs at this step (**Yes/No**).
2. If **Yes**, select a hallucination **category** from the taxonomy.
3. If **Yes**, write a short **evidence-based explanation**.

---

2) What a "step" is

In the Google Sheet:

- **Each row = one step** in a trajectory.
- Rows sharing the same `trajectory_id` belong to the same trajectory.
- Annotate steps in ascending `step_id` order whenever possible.

---

3) Links you must use (single-step + full-sequence context)

Each row includes two important URLs:

A) `render_step_url` (Column G) — Single-step view (primary)

Use this to inspect the current step (the UI screenshot/state + agent observation/reasoning/action/output shown for that step). **Your labels for this row should be based primarily on this single-step view**, with the full sequence view used for additional context when needed.

B) `render_file_url` (Column F) — Full sequence view (required for context)

This shows the **complete trajectory** and provides the full context (previous steps, UI changes, and what actually happened earlier).

Required workflow per row:

1. Open **Column G** (`render_step_url`) to inspect the current step.
2. Use **Column F** (`render_file_url`) to confirm context whenever the judgment depends on history (and this is common).
3. Fill the labels in the sheet.

---

4) Hallucination decision (Yes vs No)

4.1 Label `hallucination = Yes` when…

A hallucination occurs if there is a **verifiable mismatch** between:

- what the agent **claims** (what it says it sees/knows/did/result), or
- what the agent **does** (click/type/select/submit),

and what is supported by:

- the **GUI shown in this step**, and/or
- the **trajectory context** visible in `render_file_url`.

Common "Yes" patterns

- **Misreading the UI:** wrong text/button/price/filter state/order/list content
- **Wrong execution:** clicked the wrong target, typed into the wrong field, submitted incorrectly
- **False status claims:** "added to cart", "logged in", "filter applied", "submitted successfully" without UI evidence
- **Unsupported invention:** stating facts not present on the GUI or in the trajectory
- **Incorrect memory of prior steps:** using or claiming an incorrect remembered fact (verify using the full sequence)

4.2 Label `hallucination = No` when…

- The agent's claims/actions are consistent with the GUI and context, OR
- The step contains no meaningful claim/action (e.g., waiting/loading), OR

- You **cannot verify** a mismatch with available evidence.

4.3 If evidence is missing / unclear

If you cannot verify due to missing UI evidence, blur, or load failure:

- Set `hallucination = No`
- In `explanation`, start with **`UNCERTAIN:`** and specify what is missing.

Examples:

- `UNCERTAIN: key UI element (filter panel) is not visible; cannot verify the agent's claim.`
- `UNCERTAIN: render page failed to load (blank/404); cannot verify the action outcome.`

Important: Do not guess "Yes" without clear evidence.

---

5) Category selection (Column I)

Fill `category` only when `hallucination = Yes`. Choose the most direct root cause:

Perception

Agent **misreads** what is on the GUI (text, price, button label, selection state, ranking/order, etc.).

Execution

Execution hallucination applies when the agent's **plan/intent is correct**, but the **UI action fails to achieve the intended effect** (e.g., wrong click/type/select/submit), or the agent **claims the action succeeded** even though the UI/context shows it did not. In other words, the agent is trying to accomplish the right subgoal, but the execution does not actually complete it.

Execution (or execution error) refers to cases where the **plan/intent is correct**, but the agent **fails when translating the plan into concrete actions** (e.g., clicks/types/selects/calls the wrong thing or executes it incorrectly), leading to an outcome that deviates from the intended plan.

If the **plan is incorrect** (i.e., the chosen subgoal itself is wrong or irrelevant to the task), then it should be labeled under the category below (the "wrong plan/reasoning" category), not Execution hallucination.

Reasoning

The agent correctly reads the UI, but its reasoning or planning is logically invalid, unsupported, or suboptimal in a way that leads to incorrect decisions or outputs. This includes (1) **premature/unsupported completion** (e.g., claiming success and terminating when requirements are unmet), and (2) **incorrect planning/ordering of steps** (e.g., the correct strategy is to sort by time and then search, but the agent plans or executes the steps in the wrong order—searching first and sorting later—causing failure or incorrect results).

Memorization

Agent **remembers prior information incorrectly** (e.g., believes a filter is active, item was added, login completed, page is X but it was Y), contradicted by the GUI or the full trajectory context.

The agent incorrectly remembers prior actions or states (e.g., believes a filter is active, an item was added, login completed, or a prior step was completed/not completed), **and this is contradicted by the GUI or the full trajectory context**.

Communication

Agent's message to another agent (or another module) is incorrect relative to what actually happened / what the UI and context support.

---

6) Explanation writing (Column J)

6.1 Requirements

- **Required if `hallucination = Yes`**
- **1–3 sentences**, concise and verifiable
- Must include:
    1. the agent's claim/action, and
    2. the GUI/context evidence that contradicts it

6.2 Recommended template

```
Agent claimed/did ____, but the GUI/context shows ____ (evidence:
____). Category: ____.
```

6.3 Good vs bad explanations

Good:
`Agent said the item was added to cart, but cart count stays 0 and no confirmation appears. Category: Execution.`

Good (uses full context):
`Agent claims the "blue" filter is active, but the full sequence view shows no active filter tag in later UI states. Category: Memorization.`

Bad:
`Seems wrong.` / `Not sure.` / long text without pointing to evidence.

---

7) Google Sheet columns (what you edit)

Do **not** modify system fields. Only fill annotation fields:

- **H `hallucination` (dropdown):** must fill for every row (Yes/No)
- **I `category` (dropdown):** fill only if H=Yes
- **J `explanation` (text):** required if H=Yes; optional if H=No; use `UNCERTAIN:` if not verifiable
- **K `annotator_id`:** fill with your assigned ID (must stay consistent)

---

8) Quality checklist (per trajectory)

After finishing all steps for a `trajectory_id`:

1. **Use full context:** check `render_file_url` whenever the decision depends on earlier steps or carried-over information.
2. **Consistency:** ensure your labels across steps do not contradict.
3. **Evidence-based Yes:** every Yes must cite a clear UI/context mismatch.
4. **No guessing:** if not verifiable, mark No + `UNCERTAIN:`.
5. **Do not alter the sheet:** do not add/delete rows/columns; do not edit URLs.

---

9) Examples

Example A (Yes / Perception)

```
Agent claimed the price is $199, but the product card shows $299.
Category: Perception.
```

Example B (Yes / Execution)

```
Agent said it clicked “Add to cart”, but the UI shows no cart
confirmation and cart count did not change. Category: Execution.
```

Example C (Yes / Memorization)

```
Agent assumes the “blue” filter is active, but the full sequence view
shows no active filter tag in the UI. Category: Memorization.
```

Example D (No / Uncertain)

```
UNCERTAIN: the key UI region is not visible and page state cannot be
confirmed.
```

10) HTML Explanation

```
sites: ['classifieds']
task_id: 109
require_login: True
storage_state: ./.auth/classifieds_state.json
start_url: http://3.139.194.103:9980
geolocation: None
intent_template: How {{quantity}} does the {{item}} have?
intent: How many miles does the most expensive white sports car in the "Cars + trucks" category from Ohio have?
image: None
instantiation_dict: {'quantity': 'many miles', 'item': 'most expensive white sports car in the "Cars + trucks" category from Ohio'}
require_reset: False
eval: {'eval_types': ['string_match'], 'reference_answers': {'must_include': ['36K |OR| 36,000 |OR| 36000']}, 'reference_url': '', 'program_html': [], 'string_note': ''}
reasoning_difficulty: medium
visual_difficulty: medium
overall_difficulty: medium
comments: http://3.139.194.103:9980/index.php?page=item&id=48555
intent_template_id: 9
```

New Page

URL: http://3.139.194.103:9980/

```
[1] [IMG] [Classifieds]
[2] [A] [My account]
[3] [A] [Logout]
[4] [A] [Publish Ad]
[] [StaticText] [What are you looking for today?]
[5] [INPUT] []
[6] [SELECT] [Select a categoryAntiquesAppliancesArts + craftsAuto partsBeauty + healthBikesBoatsBooksCars + trucksCell phonesCollectiblesComputer partsComputersElectronicsFarm +
gardenFurnitureHouseholdJewelryMoto]
[7] [BUTTON] [Search]
[] [StaticText] [Latest Listings]
[] [StaticText] [JBL Powered PA Speaker w/ Stand                    150.00 $                          Music instruments             Borough of Red Lion  (Pennsylvania)
2023/11/16                    ]
[8] [IMG] [JBL Powered PA Speaker w/ Stand]
[9] [A] [JBL Powered PA Speaker w/ Stand]
[] [StaticText] [150.00 $]
[] [StaticText] [xbox series x / with extras                      350.00 $                          Video gaming              Pennwyn  (Pennsylvania)              2023/11/16
Used xbox s]
[10] [IMG] [xbox series x / with extras]
[11] [A] [xbox series x / with extras]
[] [StaticText] [350.00 $]
[] [StaticText] [Canon EF 100-400mm f/4.5-5.6L IS II USM with extender EF 1.4X III                    1645.00 $                         Photo + video             College Park
(Maryland)               2023/11/16  ]
[12] [IMG] [Canon EF 100-400mm f/4.5-5.6L IS II USM with extender EF 1.4X III]
```

`render_file_url` (Full Trajectory View)

`render_file_url` is a visualization of the **entire trajectory** (all steps in sequence).
Use this page to understand the **full context**, including what happened in earlier steps, how the

UI state evolved, and whether the agent's later claims/actions are consistent with the trajectory history.

`intent` (Current Task / User Question)

`intent` describes the **current task the agent is trying to solve**, usually phrased as a user question.
Example:
`How many miles does the most expensive white sports car in the "Cars + trucks" category from Ohio have?`

This is the goal the agent should work toward during the trajectory.

`eval` (Ground-Truth / Evaluation Target)

`eval` contains the **expected answer requirements** for the task, used for evaluation.
It may include:

- the evaluation type (e.g., string match), and
- reference answer constraints (e.g., keywords or numeric formats that must appear).

Example:
`{'eval_types': ['string_match'], 'reference_answers': {'must_include': ['36K |OR| 36,000 |OR| 36000']}, ...}`

Interpretation: the final answer should match one of the accepted forms (e.g., "36K", "36,000", or "36000").

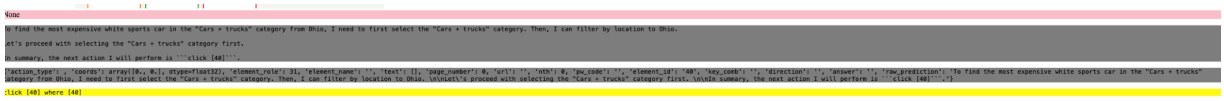
`render_step_url` / "New page" (Current Step View)

The "New page" (i.e., `render_step_url`) shows the **current step only**: the current UI/page state plus the agent's associated **thinking** and **action** for that step.

This page corresponds to the **current row** in the sheet — you should annotate hallucination/category/explanation **for this step** based primarily on what is shown here, using the full trajectory view (`render_file_url`) when additional context is needed.

Color-coded UI Elements (How to Read Each Step)

On the step visualization page, different background colors indicate different parts of the step. Please interpret them as follows:



1) Pink background = Previous step's action

- The text highlighted with a **pink background** shows the **action taken in the previous step**.
- Therefore, for the **first "New page"** in a trajectory, the previous action is **None** (because there is no earlier step).

2) First gray box = Agent's reasoning / thought (the key part for annotation)

- The **first gray background block** contains the agent's **thinking/reasoning** for the current step.
- **This is one of the main sources you should use for hallucination annotation**, together with the UI evidence.

3) Second gray box = Parsed/structured output (ignore)

- The **second gray background block** is a structured result **parsed from the first gray block**.
- You **do not need to annotate or rely on this block**. Please focus on the first gray block instead.

4) Yellow background = Current step's action to execute

- The text highlighted with a **yellow background** is the **action the agent intends to perform in the current step**.

- The subsequent **“New page”** corresponds to the UI state **after executing this yellow action**.

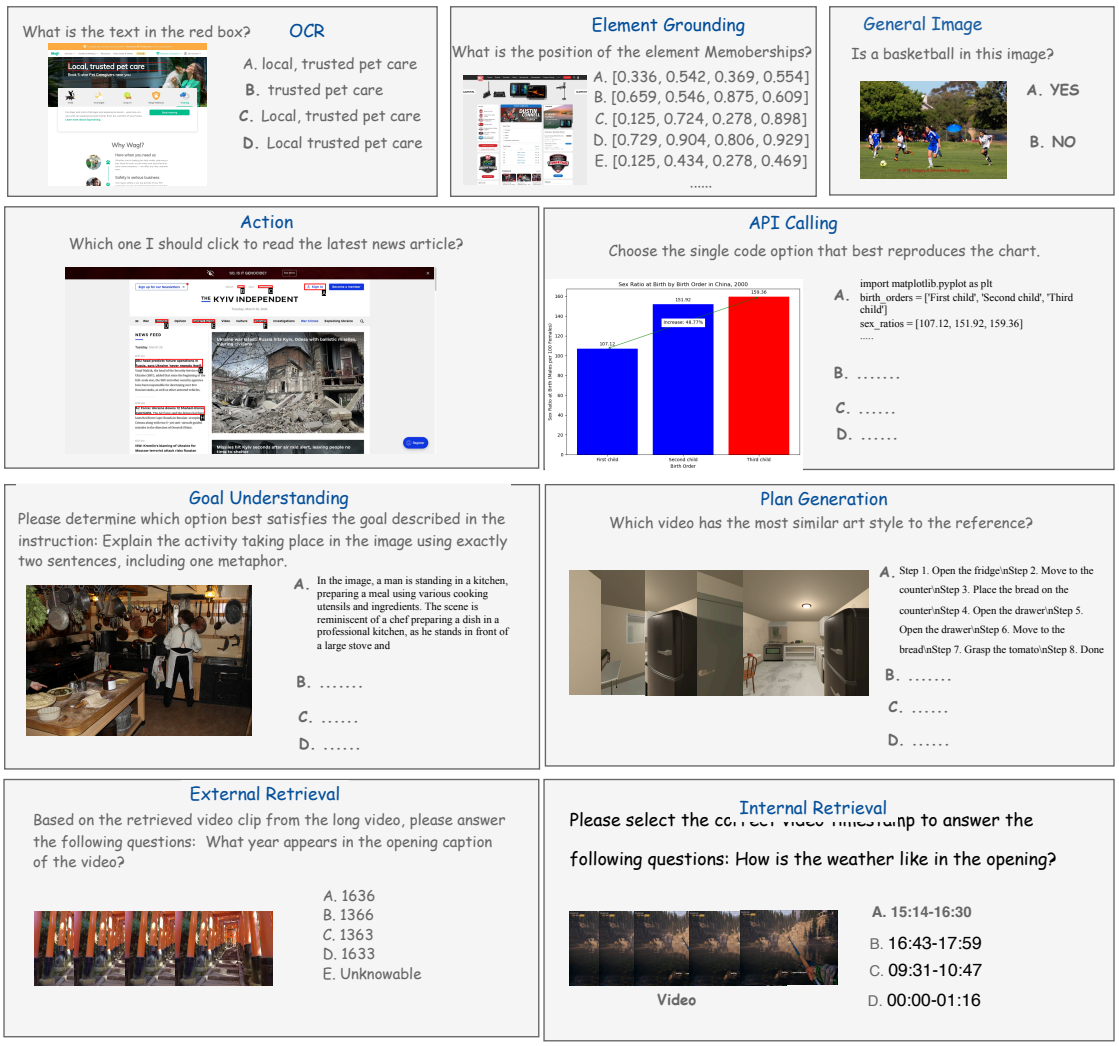


Figure 4: Overview of representative task types in HALO-Static.

Figure 5: Overview of representative task types in HALO-Static.