AGENTVISTA: Evaluating Multimodal Agents in Ultra-Challenging Realistic Visual Scenarios

Zhaochen Su[1*] **Jincheng Gao**[1] **Hangyu Guo**[1] **Zhenhua Liu**[1] **Lueyang Zhang**[1]
Xinyu Geng[1] **Shijue Huang**[1] **Peng Xia**[2] **Guanyu Jiang**[3] **Cheng Wang**[4]
Yue Zhang[1] **Yi R. (May) Fung**[1*] **Junxian He**[1*]
[1]Hong Kong University of Science and Technology
[2]University of North Carolina at Chapel Hill
[3]Zhejiang University [4]National University of Singapore
Project page: agentvista-bench.github.io
Code: github.com/hkust-nlp/AgentVista

Abstract

Real-world multimodal agents solve multi-step workflows grounded in visual evidence. For example, an agent can troubleshoot a device by linking a wiring photo to a schematic and validating the fix with online documentation, or plan a trip by interpreting a transit map and checking schedules under routing constraints. However, existing multimodal benchmarks mainly evaluate single-turn visual reasoning or specific tool skills, and they do not fully capture the realism, visual subtlety, and long-horizon tool use that practical agents require. We introduce AGENTVISTA, a benchmark for generalist multimodal agents that spans 25 sub-domains across 7 categories, pairing realistic and detail-rich visual scenarios with natural hybrid tool use. Tasks require long-horizon tool interactions across modalities, including web search, image search, page navigation, and code-based operations for both image processing and general programming. Comprehensive evaluation of state-of-the-art models exposes significant gaps in their ability to carry out long-horizon multimodal tool use. Even the best model in our evaluation, GPT-5.6-SOL with tools, achieves only 35.41% overall accuracy, and hard instances can extend to 30 tool-calling turns. We expect AGENTVISTA to accelerate the development of more capable and reliable multimodal agents for realistic and ultra-challenging problem solving.

1 Introduction

Humans seamlessly integrate multi-sensory information to tackle complex real-world problems [Stein, 2012]. With the rapid evolution of AI agents [Wang et al., 2024a, Comanici et al., 2025, OpenAI, 2025e, Team et al., 2026], developing visual agentic intelligence becomes essential. For instance, an agent is expected to assist in shopping by scanning shelf products and retrieving nutritional information to satisfy user health constraints, or support troubleshooting by linking malfunction photos with schematic diagrams to diagnose specific faults. However, a major challenge in developing such multimodal agents is the absence of a benchmark based on realistic scenarios that covers the diversity and complexity of long-horizon tool interactions across different modalities, which limits reliable evaluation of agent capabilities in open domains [Xie et al., 2024, Li et al., 2025a].

*Corresponding authors: Zhaochen Su (zsubf@connect.ust.hk), Yi R. (May) Fung (yrfung@ust.hk), and Junxian He (junxianh@cse.ust.hk).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

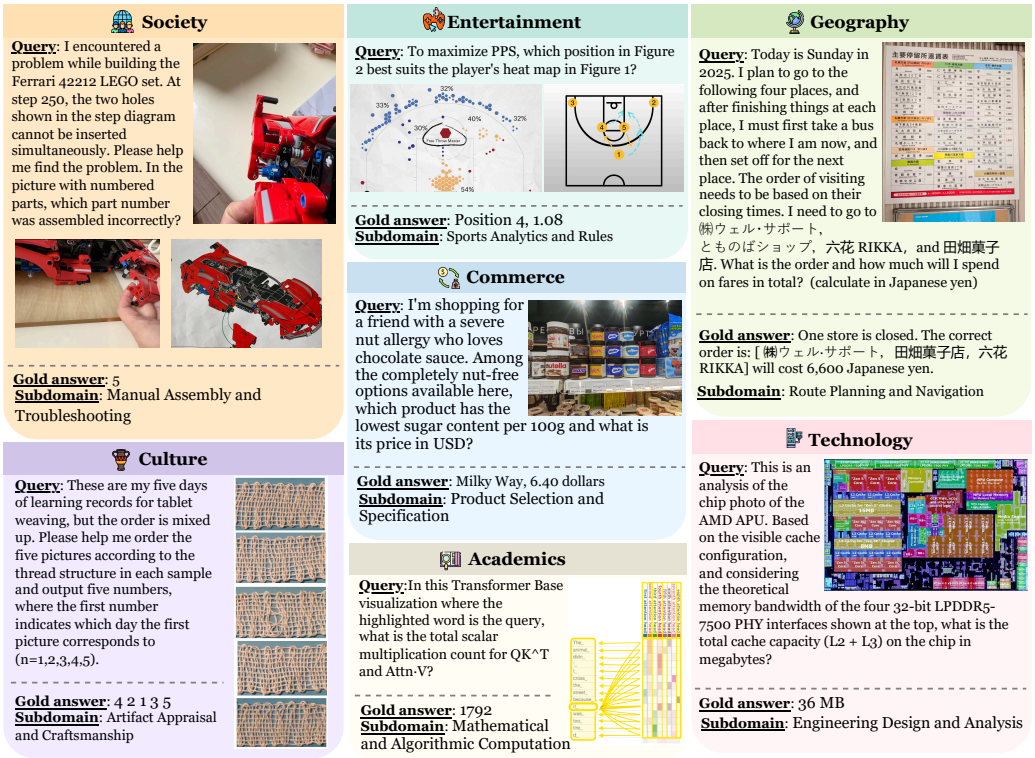


Figure 2: Sampled AGENTVISTA examples from each domain. Each query is grounded in complex, real-world visual scenes and is designed to elicit agentic tool use with multi-step reasoning toward a unique, verifiable answer.



Figure 1: A representative AGENTVISTA task: matching flooring styles across rooms, retrieving product specs, and computing the final cost via interleaved tool use.

Traditional multimodal benchmarks [Antol et al., 2015, Hudson and Manning, 2019, Yue et al., 2024a, Wang et al., 2024b, Scale AI, 2025] focus on assessing visual perception and complex reasoning capabilities. Recently, a growing number of benchmarks have emerged to evaluate multimodal agentic behaviors [Ma et al., 2024, Li et al., 2025b, Ashraf et al., 2025, Guo et al., 2025, Tao et al., 2025, Geng et al., 2025]. However, these evaluations typically present two main gaps: ❶ **Capability-Specific Evaluation**: They typically emphasize particular capabilities, focusing on skills such as visual manipulation [Wang et al., 2025, Lai et al., 2025], web browsing [Li et al., 2025c, Tao et al., 2025], or code generation [Yang et al., 2024]. This narrow focus makes it difficult to evaluate generalist agents that must combine multiple skills and remain reliable in long-horizon workflows.

❷ **Trade-off between Realism and Difficulty**: Practical agent tasks are difficult because they combine cluttered visual evidence with long-horizon tool use under constraints. Yet many benchmarks increase difficulty by simplifying the visual state or by relying on tool patterns that deviate from everyday workflows, which can shift the bottleneck away from realistic grounding and interaction. For example, VisualToolBench pre-processes the input images to facilitate specific visual operations [Guo et al., 2025]. While this design is effective for evaluating visual manipulation, it also shifts the problem from reasoning over natural visual states to operating on curated inputs.

To address these gaps, we introduce AGENTVISTA, a benchmark designed to evaluate generalist multimodal agents on diverse, realistic, and challenging tasks. AGENTVISTA contains 209 tasks spanning 25 sub-domains across 7 categories, including commerce, geography, society, technology, entertainment, culture, and academics, and grounds each query in detail-rich visual states such as daily photos, screenshots, and technical diagrams, with both single-image and multi-image inputs.

Table 1: Comparison with representative multimodal agent benchmarks. Operation abbreviations: **VO.** (Visual Operations), **VS.** (Visual Search), **TS.** (Text Search), and **CE.** (Code Execution). "# Turns" reports the average number of tool-calling turns by GPT-5, used as a proxy for task complexity. AGENTVISTA jointly covers all four tool operations and multi-image inputs, with markedly longer interaction trajectories.

BENCHMARK	VO.	VS.	TS.	CE.	MULTI IMAGE	# Turns
MMSEARCH-PLUS	✗	✓	✓	✗	✓	4.60
BROWSECOMP-VL	✗	✓	✓	✓	✗	4.30
VISUALTOOLBENCH	✓	✗	✓	✓	✗	4.46
AGENTVISTA (Ours)	✓	✓	✓	✓	✓	12.67

Table 2: Summary statistics of the AGENTVISTA benchmark. Lengths are measured in characters.

STATISTIC	NUMBER
Total queries	209
Total images	307
Primary categories	7
Secondary categories	25
Avg. query length	408.2
Avg. answer length	42.0
Image distribution	
– Single-image	152 (72.7%)
– Multi-image	57 (27.3%)

Each query is manually authored to reflect authentic user intent and is subjected to strict quality control, where every instance is carefully reviewed to ensure mandatory visual dependency and a unique, verifiable answer. Every task requires long-horizon interaction with interleaved tools, where the agent repeatedly grounds visual cues, retrieves external information, and verifies intermediate decisions. Table 1 summarizes the key differences between AGENTVISTA and representative agentic multimodal benchmarks. Figure 1 shows a representative example from AGENTVISTA motivated by a real home renovation need: the agent needs to match flooring styles across scenes, verify the target room with image-based checks, retrieve product specifications online, and compute a deterministic final cost from the room size and packaging information.

AGENTVISTA is evaluated in a controlled yet practical setting, where we adopt four widely used tools that cover the core interaction patterns of real-world multimodal agents, including web search, image search, page navigation, and code-based operations for both image processing and general programming. Our experiments on representative open-source and commercial MLLMs show that AGENTVISTA remains far from being solved, leaving substantial room for improvement. Even the best performance in our evaluation, GPT-5.6-SOL with tools, achieves only 35.41% overall accuracy. Further error analysis shows that many failures start with visual misidentification and then lead to wrong retrieval and unreliable tool use over many steps. To facilitate future research, we release both the AGENTVISTA benchmark and a lightweight yet general agent framework to facilitate reproducible evaluation and accelerate progress on long-horizon multimodal tool use.

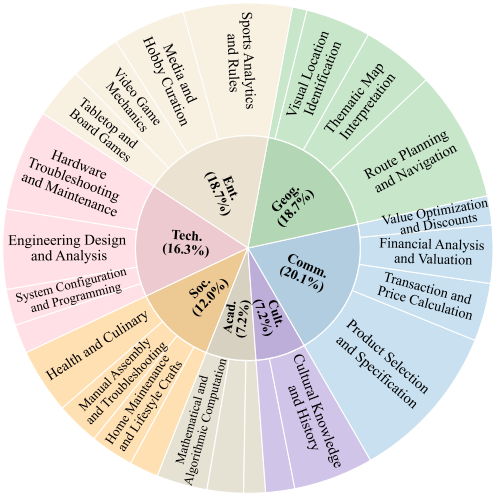


Figure 3: The categorization of AGENTVISTA. The benchmark spans 7 major categories and 25 sub-domains, covering a broad range of realistic and challenging multimodal agent scenarios.

2 The AGENTVISTA Benchmark

2.1 Overview of AGENTVISTA

We introduce AGENTVISTA, a benchmark for evaluating generalist multimodal agents on realistic and ultra-challenging tasks. AGENTVISTA focuses on realistic user requests that are still hard in practice and require long-horizon tool use grounded in visual evidence. AGENTVISTA contains 209 tasks spanning 25 sub-domains across 7 categories: Technology, Commerce, Geography, Entertainment, Society, Academics, and Culture. Each instance takes about four hours of expert effort to construct and curate, and the overall scale is comparable to recent agentic benchmarks [Li et al., 2025a,c].

The domain distribution and dataset composition are summarized in Table 2 and Figure 3. As shown in Figure 2, tasks are built from authentic user needs and require multi-step reasoning with tool use.

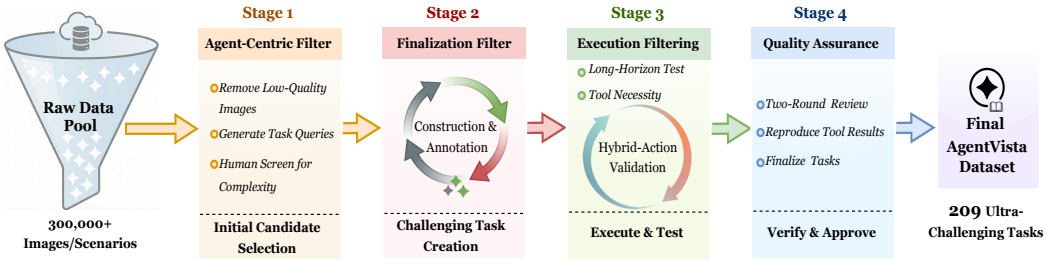


Figure 4: Overview of the AGENTVISTA dataset construction pipeline, consisting of agent-centric filtering, expert finalization, execution filtering, and two-round verification to produce realistic and ultra-challenging multimodal agent tasks.

For example, an agent may need to read key constraints from a photo or screenshot, retrieve missing details from external resources, and then combine multiple pieces of evidence to produce the final answer. This includes diagnosing a hardware issue by matching visible components to technical documentation, selecting a product that satisfies allergy and nutrition constraints by comparing labels with online specifications, and planning a route under time and transit limits by reading schedules from images and verifying them with web search. To enable robust and scalable evaluation, each instance is paired with a clear and deterministic ground truth answer, typically a short phrase or a numeric value.

2.2 Data Construction

2.2.1 Core Design Principles

We design AGENTVISTA based on three principles:

- **Vision-centric tasks with realistic images.** Each task requires obtaining the key evidence from the visual input. The images are real and contain visual details to support visual understanding, such as small but important cues, multiple related objects, or subtle differences across views. The query avoids stating the key information in text and avoids questions that can be answered by a keyword search. These constraints ensure that solving the task relies on understanding and comparison of visual details, rather than on textual shortcuts.
- **Natural interleaved hybrid tool use.** Each task requires using different tool types together, and the interaction must include interleaved tool calls across at least two tool categories. The intended solution should mix visual tools and text-based tools, such as using image search or image processing to gather visual evidence, then using web search or page navigation to retrieve needed facts, and finally combining the evidence to reach the answer. Tool use must follow natural and real-world workflows. Each tool call should be necessary for solving the task, rather than added only to make the interaction longer. To keep tasks realistic and challenging, we favor instances that require grounding tool outputs in the visual input under explicit constraints.
- **Easy to verify and stable over time.** Following recent evaluation protocols [Li et al., 2025c, Wei et al., 2025], each task has a concise target answer in a fixed format, such as a number, an entity name, or a short description. This design makes the evaluation process simple and accurate, similar to math tasks. Additionally, we address the issue of information changing over time. Annotators verify facts against reliable sources. When necessary, we include specific time constraints in the question to ensure the ground truth remains valid.

2.2.2 Dataset Creation Pipeline

We build AGENTVISTA from 300k+ real images and real user needs collected from public model arenas, annotator-captured daily scenarios, and private community forums, with details in Appendix A.2. The dataset construction pipeline is shown in Figure 4.

Stage 1: Agent-centric filtering. We start with model-assisted mining and filtering to identify candidate initial states that reflect realistic daily workflows. We first use CLAUDE-OPUS-4 to filter the raw image pool by removing cases with limited visual information or weak agentic potential,

such as pure OCR screenshots, single-object landmark photos, and images that can be solved without meaningful visual reasoning. We provide CLAUDE-OPUS-4 with our tool schema and ask it to propose an initial task query that is compatible with the available tools and has a verifiable answer format. The proposed query serves as a candidate starting point for downstream curation. We then apply human screening to retain only images with sufficiently rich visual evidence and queries that support a natural task formulation with hybrid tool use. To avoid simple cases, we prioritize candidates with non-trivial constraints and keep only those that naturally require multi-step reasoning rather than a single direct lookup.

Stage 2: Expert finalization. We recruit and train expert annotators on the project scope, taxonomy, and quality requirements, and ask them to finalize each candidate produced in Stage 1. Starting from the image and the initial query, annotators rewrite the query into a realistic user request while keeping it self-contained and vision-centric. Realism is enforced by preserving the original visual state and intent, and by expressing constraints in the way users typically specify them, such as time, budget, compatibility, and safety requirements. To make tasks ultra-challenging in a natural way, annotators select cases where the answer depends on fine-grained visual cues and cannot be obtained by a single direct lookup. They ensure that solving the task requires combining visual evidence with information gathered from tools, and that the process includes necessary interleaving across tool types. Annotators then produce a deterministic target answer and record the key evidence and tool steps used to obtain it, which enables later checking.

Stage 3: Execution filtering. We validate each instance by executing the candidate task in our tool environment and checking that the annotated answer is supported by reproducible tool outputs. During this process, we run GEMINI-3-FLASH in the same tool environment to screen for tool-use diversity, and we retain only tasks that require interleaved calls across at least two categories. Furthermore, we run GEMINI-2.5-PRO with tool access disabled and remove samples that can be solved from the prompt alone.

Stage 4: Two-round verification. Finally, we conduct two rounds of verification. The first round removes instances with insufficient visual evidence, weak visual dependency, or questionable answer validity. In the second round, a separate group re-checks each instance by following the evidence and tool steps recorded by annotators, and confirms that the final answer is supported by the visual cues and the tool outputs. Instances with unclear evidence, unstable answers, or unrealistic workflows are removed. The remaining instances form the final AGENTVISTA benchmark.

Filtering statistics. We begin with 300k+ candidate images. Stage 1 uses model-assisted filtering and human screening to select 568 potential initial states, 0.19% of the raw pool. Stage 2 expert finalization yields 315 tasks after rewriting the initial queries into realistic user requests and adding deterministic target answers. Stage 3 execution filtering retains 241 tasks by validating reproducible tool outputs, enforcing interleaved calls across at least two tool categories, and removing tasks solvable when tool access is disabled. Stage 4 two-round verification selects the final 209 tasks by re-checking visual evidence, recorded tool steps, and answer validity. On average, constructing a single instance takes about 4 hours, and expert annotators take about 30 minutes to solve an instance.

2.2.3 Tool environment

AGENTVISTA supports a compact set of tools that cover common multimodal agent workflows. Models can call `web_search` to retrieve web pages, `visit` to open and navigate a page, and `image_search` to locate images when a query requires external visual references. We also provide `code_interpreter`, which supports both programming and image processing. It enables arithmetic and parsing, structured extraction, and operations such as cropping, resizing, measuring, and comparing visual regions when needed. All tools are exposed with detailed descriptions and structured inputs and outputs, so the model can decide when to call a tool and how to use the returned results. Detailed tool definitions are provided in Appendix B.1.

Table 3: Main results on our proposed AGENTVISTA. Domain abbreviations: **COMM.** (Commerce), **GEOG.** (Geography), **ENT.** (Entertainment), **TECH.** (Technology), **SOC.** (Society), **ACAD.** (Academics), and **CULT.** (Culture). Input mode abbreviations: **SINGLE.** (Single-image) and **MULTI.** (Multi-image). The best model in each column is in **bold**; the second best is underlined. Overall, GPT-5.6-SOL achieves the highest accuracy among all evaluated models.

MODEL	BY CATEGORY							By Input Mode		Summary	
	COMM.	GEOG.	ENT.	TECH.	SOC.	ACAD.	CULT.	SINGLE.	MULTI.	OVERALL	# TURNS
GROK-4	11.90	23.08	10.26	20.59	28.00	0.00	0.00	14.47	17.54	15.31	16.44
CLAUDE-SONNET-4	9.52	17.95	5.13	29.41	16.00	20.00	6.67	11.84	22.81	14.83	5.37
CLAUDE-OPUS-4	19.05	15.38	7.69	26.47	24.00	20.00	6.67	12.50	29.82	17.22	6.89
CLAUDE-OPUS-4.1	11.90	23.08	12.82	29.41	20.00	26.67	13.33	17.11	24.56	19.14	7.28
CLAUDE-SONNET-4.5	11.90	23.08	10.26	26.47	24.00	20.00	13.33	17.76	19.30	18.18	9.99
CLAUDE-OPUS-4.6	21.43	23.08	**28.21**	47.06	36.00	33.33	26.67	28.29	35.09	30.14	7.75
CLAUDE-SONNET-4.6	19.05	25.64	10.26	20.59	44.00	40.00	26.67	22.37	28.07	23.92	12.91
GEMINI-3-FLASH	16.67	20.51	12.82	29.41	28.00	40.00	20.00	19.08	29.82	22.01	7.78
GEMINI-3-PRO	16.67	28.21	23.08	32.35	32.00	40.00	40.00	23.68	38.60	27.75	6.67
GEMINI-3.1-PRO	21.43	25.64	23.08	44.12	28.00	33.33	40.00	25.66	38.60	29.19	6.09
O3	21.43	17.95	10.26	23.53	44.00	26.67	13.33	19.08	28.07	21.53	13.18
O4-MINI	2.38	10.26	5.13	8.82	12.00	13.33	0.00	7.24	7.02	7.18	1.89
GPT-5	23.81	23.08	15.38	35.29	28.00	26.67	26.67	24.34	26.32	24.88	12.67
GPT-5.1	23.81	15.38	17.95	26.47	24.00	40.00	40.00	20.39	33.33	23.92	17.14
GPT-5.2	21.43	20.51	23.08	38.24	28.00	33.33	20.00	24.34	29.82	25.84	13.85
GPT-5.4	**30.95**	**30.77**	15.38	52.94	40.00	33.33	26.67	30.26	38.60	32.54	3.68
GPT-5.6-SOL	23.81	23.08	23.08	50.00	48.00	**66.67**	**46.67**	**32.89**	**42.11**	**35.41**	5.48
GPT-6.1-SOL	23.81	23.08	20.51	**55.88**	**52.00**	46.67	40.00	32.24	40.35	34.45	5.18

3 Experiments

3.1 Experimental Setup

Models. We evaluate a broad set of frontier multimodal models that are commonly used as generalist agents. Specifically, we test O3, O4-MINI [OpenAI, 2025a], GPT-5 [OpenAI, 2025b], GPT-5.1 [OpenAI, 2025c], GPT-5.2 [OpenAI, 2025d], GPT-5.4 [OpenAI, 2026a], GEMINI-3-FLASH [Google DeepMind, 2025a], GEMINI-3-PRO [Google DeepMind, 2025b], GEMINI-3.1-PRO [Google DeepMind, 2026], GROK-4 [xAI, 2025], CLAUDE-SONNET-4 [Anthropic, 2025b], CLAUDE-OPUS-4 [Anthropic, 2025b], CLAUDE-OPUS-4.1 [Anthropic, 2025a], CLAUDE-SONNET-4.5 [Anthropic, 2025c], CLAUDE-OPUS-4.6 [Anthropic, 2026a], CLAUDE-SONNET-4.6 [Anthropic, 2026b], GPT-5.6-SOL [OpenAI, 2026b], and GPT-6.1-SOL [OpenAI, 2026c].

Evaluation Setup. For all experiments, we use a temperature of 0.6 and cap the tool interaction budget at 30 turns for every model. Since AGENTVISTA provides concise target answers in deterministic formats, evaluation reduces to verifying the final answer. We use GPT-5.6-LUNA as a fixed judge model to assess whether a model's final response matches the annotated ground truth under the required format. We report accuracy as the evaluation metric.

3.2 Main Results

We report the overall performance in Table 3. We make the below three observations.

AGENTVISTA is ultra-challenging. The results show that AGENTVISTA remains difficult for current multimodal agents. Even the best-performing model, GPT-5.6-SOL, achieves only 35.41% overall accuracy, indicating substantial headroom. Performance is also low for a large portion of models: 2 out of 18 models score below 15% overall accuracy. The average number of turns further reflects this difficulty: GPT-5.1 and GROK-4 reach the longest trajectories at 17.14 and 16.44 turns on average, while GPT-5.2, O3, CLAUDE-SONNET-4.6, and GPT-5 all exceed 12 turns. GPT-5.6-SOL reaches the highest score with 5.48 turns on average, and GPT-5.4 remains strong with a lower average of 3.68 turns, but this reflects adaptive tool-routing rather than reduced task difficulty: on harder cases, their trajectories still extend to 25–30 turns of interleaved tool use, matching the depth seen across other top models. GPT-6.1-SOL trails GPT-5.6-SOL mainly on ACADEMICS and ENTERTAINMENT, and their 0.96-point overall gap lies within decoding and judging noise. We also observe a sizable gap between open-source and closed-source models, suggesting substantial room for open-source multimodal agents. We further verify in Appendix B.2 that the accuracy estimate is stable and low-variance under random sub-sampling. We report additional open-source baselines in Appendix B.3 and further analyze failure patterns in Section 4.3.

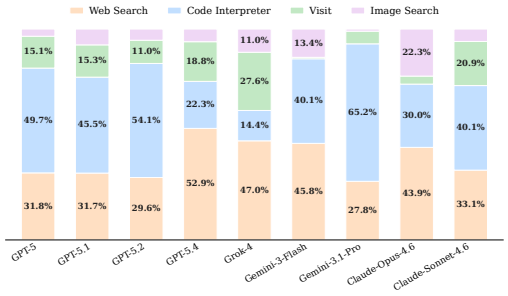


Figure 5: Tool-use distribution across selected multimodal models, showing distinct preferences between code execution and web search.

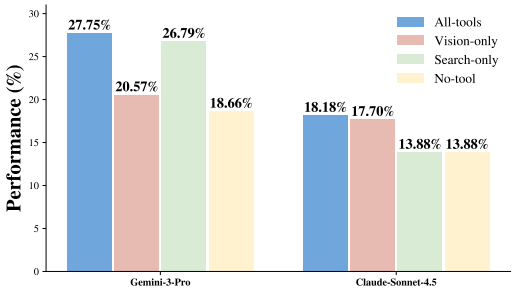


Figure 6: Tool ablation on GEMINI-3-PRO and CLAUDE-SONNET-4.5. Both models perform best with the full tool suite.

Domain strengths differ across model families. Performance varies noticeably across categories, revealing complementary strengths among model series. GPT-6.1-SOL leads TECHNOLOGY and SOCIETY, and GPT-5.6-SOL leads ACADEMICS and CULTURE, both with clear margins. The GPT-5 family shows strong coverage on practical categories, with GPT-5.4 leading COMMERCE and GEOGRAPHY and placing second on TECHNOLOGY. The Claude series is comparatively stronger on categories that emphasize careful reading and constraint following: CLAUDE-OPUS-4.6 leads ENTERTAINMENT, with four models tying for second place. The Gemini series remains strong on categories that reward confident visual grounding: GEMINI-3-PRO places second on GEOGRAPHY and stays in the leading group on ENTERTAINMENT, ACADEMICS, and CULTURE, demonstrating consistent strength across most categories.

Multi-image inputs are not uniformly harder than single-image inputs. For nearly all evaluated models, accuracy with multi-image inputs is higher than with single-image inputs, and the gap is sometimes sizeable. The gain is largest for CLAUDE-OPUS-4 at +17.3 points, improving from 12.50% to 29.82%, followed by GEMINI-3-PRO at +14.9 points, from 23.68% to 38.60%, and GPT-5.1 at +12.9 points, from 20.39% to 33.33%. This pattern matches how our multi-image instances are constructed. Additional views often provide complementary evidence, reduce ambiguity, and reveal details that are missing in a single shot, which can make grounding and downstream retrieval more reliable. The two newest models follow the same pattern. GPT-5.6-SOL improves from 32.89% to 42.11% and GPT-6.1-SOL improves from 32.24% to 40.35% when moving from single-image to multi-image inputs. While multi-image inputs still require cross-image alignment, the results suggest that the main bottleneck remains long-horizon tool use and constraint tracking, rather than the presence of multiple images itself.

4 Further Analysis

4.1 Tool Distribution Analysis

In this section, we analyze the distribution of tool calls across models. As shown in Figure 5, the GPT-5 series and GEMINI-3.1-PRO rely most heavily on the code interpreter. We further break down code interpreter calls by operation type in Appendix B.4. The results suggest that these models more often perform image-centric operations during problem solving, such as zooming in, cropping, resizing, and measuring regions. For most models, *crop* is the most frequent operation, indicating that many trajectories depend on localized visual grounding before proceeding to retrieval or computation. Second, GROK-4, GEMINI-3-FLASH, and the CLAUDE series call web search most often, indicating a stronger preference for retrieval-driven workflows. Across all models, image search is used less frequently than the other tools. In the next tool ablation study, we quantify how each tool contributes to performance and how accuracy changes when a tool is removed.

4.2 Tool Ablation Study

In this section, we ablate tool access to quantify how each tool modality contributes to performance.

Experimental setup. We evaluate three settings with prompts lightly adapted to reflect the available capabilities, while keeping the evaluation protocol and inference hyperparameters fixed. ❶ **Vision only**: the agent has access only to a visual manipulation environment, enabling image processing operations for inspection and transformation, but no external retrieval. ❷ **Search only**: the agent can retrieve external evidence through both image-based and text-based search, and can read retrieved webpages, but cannot perform tool-based visual manipulation or programmatic verification. ❸ **No tool**: the agent relies purely on direct generation without any tool assistance.

Key findings. Figure 6 shows that using the full tool suite yields the best performance for both models, confirming that AGENTVISTA rewards hybrid workflows that combine visual manipulation and retrieval. For GEMINI-3-PRO, the full tool setting reaches 27.75% accuracy, higher than the vision-only setting at 20.57% and the no-tool setting at 18.66%. For CLAUDE-SONNET-4.5, the full tool setting achieves 18.18%, slightly above the vision-only setting at 17.70%, while the search-only and no-tool settings both drop to 13.88%. We also find that the role of retrieval differs across models. For GEMINI-3-PRO, the search-only setting reaches 26.79%, close to the full tool setting. This suggests that its strong visual perception enables it to extract reliable cues from images and benefit primarily from retrieval and page navigation, while visual manipulation mainly supports inspection and verification. In contrast, CLAUDE-SONNET-4.5 relies more on visual manipulation than retrieval, since the vision-only setting remains close to the full tool setting, whereas the search-only setting degrades substantially.

4.3 Error Analysis

To understand the main bottlenecks on AGENTVISTA, we analyze failures from three representative frontier models, GPT-5.4, GEMINI-3.1-PRO, and CLAUDE-OPUS-4.6. For each incorrect case, we assign an error label, including tool execution failure, visual misidentification, knowledge hallucination, calculation error, instruction misinterpretation, and others. The labels are generated by GEMINI-3-FLASH based on the model trajectories, and the distributions are shown in Figure 7. Detailed definitions for each error type are provided in Appendix C. Figure 7 shows a clear trend that visual misidentification is the main failure mode across all three models. This aligns with the design of AGENTVISTA, where tasks are grounded in realistic and cluttered visual states and often depend on small but critical details. From bad cases, we find that frontier agents can often zoom in to the relevant region, but they still fail when the image is blurry or the key cue is visually subtle. Knowledge hallucination is the second most common error type, which also matches our benchmark design. Many tasks require applying diverse world knowledge to long-horizon tool interactions, and current models still struggle to resolve long-tail facts reliably even with web search. We include representative good and bad cases with detailed explanations in Appendix E. Overall, these results suggest that AGENTVISTA can expose practical weaknesses in both fine-grained visual understanding and knowledge-grounded reasoning under realistic tool use.

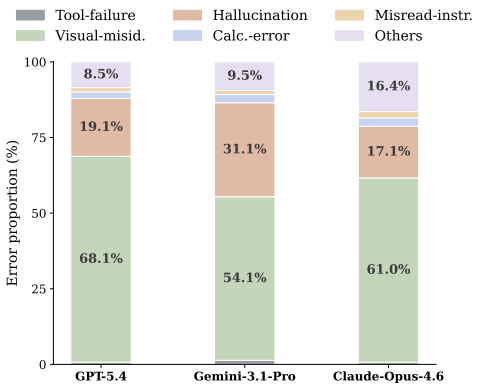


Figure 7: Error category distribution on AGENTVISTA across three multimodal models. *Visual misidentification* is the dominant failure mode, indicating errors often originate from incorrect visual grounding.

4.4 Test-Time Scaling

To study whether additional sampling at test time can improve performance on AGENTVISTA, we evaluate test-time scaling on GEMINI-3-FLASH. We generate K solutions per instance and use GEMINI-3-FLASH as the reward model to select a final answer when selection is required. We follow the same evaluation protocol as in prior experiments. Table 4 reports three settings: RANDOM1@K, which randomly selects one of the K samples as a lower bound, Best-of-K (BON@K), which selects the highest-scoring sample under the reward model, and PASS@K, which measures whether at least one of the K samples is correct as an upper bound.

Key findings. Table 4 shows that test-time scaling consistently improves performance. Under BoN, accuracy increases from 22.01% at K=1 to 31.58% at K=16. The upper bound rises even more, with Pass@K increasing from 22.01% at K=1 to 52.63% at K=16. In contrast, Random1@K remains low and does not improve with larger K, indicating that gains mainly come from better selection rather than sampling alone. Despite these improvements, scaling alone is not sufficient to solve AgentVista. Even at K=16, BoN reaches only 31.58%, while Pass@16 is 52.63%. For reference, the best-of-16 result still trails a single standard run of GPT-5.6-Sol (35.41%). This gap indicates substantial room for reinforcement learning or other optimization methods that can better close the gap between selection and the achievable upper bound, and more broadly highlights the need for stronger long-horizon tool use and more reliable visual grounding.

Table 4: Test-time scaling results under different sampling budgets K on Gemini-3-Flash. We report Random1@K as a lower bound, Best-of-K (BoN@K) selected by a reward model, and Pass@K as an upper bound. All values are accuracies in %.

Setting	K=1	K=2	K=4	K=8	K=16
Random1@K	22.01	20.07	19.19	18.05	19.01
BoN@K	22.01	25.84	27.27	29.19	31.58
Pass@K	22.01	27.03	35.18	43.54	52.63

5 Related Work

5.1 Multimodal Agents and Tool Use

Recent years have witnessed rapid progress in large multimodal models that combine visual perception with language-based reasoning [Peng et al., 2023, Liu et al., 2023, Zhu et al., 2023, Li et al., 2023]. A key step toward practical multimodal agents is to couple these models with tools so they can inspect visual evidence, verify intermediate hypotheses, and refine solutions over multiple steps. OpenAI o3 and o4-mini follow this direction by manipulating user-provided images during reasoning through operations such as cropping, zooming, and rotation, and coordinating these visual operations with other tools when needed [OpenAI, 2025e]. This paradigm has inspired open systems that study tool-driven multimodal reasoning and long-horizon interaction [Su et al., 2025a,b]. Recent work also explores stronger training signals for repeated grounding, such as reinforcement learning for interleaved perception and reasoning [Zheng et al., 2025], and extends multimodal agents with web and code tools for mixed tool use in realistic settings [Hong et al., 2025, Geng et al., 2025, Jiang et al., 2026]. Despite this progress, there is still no benchmark that evaluates generalist multimodal agents on realistic, ultra-challenging tasks. AgentVista fills this gap by focusing on long-horizon, interleaved tool use grounded in real visual inputs.

5.2 Multimodal Agent Benchmarks

Early multimodal benchmarks mainly evaluate perception and visual reasoning in static question answering, where models respond from a fixed image and text context without interaction [Antol et al., 2015, Hudson and Manning, 2019, Lu et al., 2023, Yue et al., 2024b, Wang et al., 2024b]. While useful, they do not test whether an agent can choose actions, call tools, and verify intermediate results. Recent agent benchmarks add tool use, including multi-step planning [Ma et al., 2024], web browsing and search [Li et al., 2025c, Tao et al., 2025], and tool-assisted visual reasoning and active perception [Wu and Xie, 2024, Lai et al., 2025, Li et al., 2025b, Ashraf et al., 2025]. More recent works further move toward interleaved tool settings, but the visual evidence is often relatively clean or lightweight, which makes perception less demanding, and the resulting tool trajectories tend to be shorter and less diverse [Guo et al., 2025, Hong et al., 2025, Chen et al., 2026]. AgentVista addresses this gap by emphasizing realistic visual inputs and long-horizon workflows that require repeated visual checking and interleaved use of multiple tool types.

6 Discussion

Limitations. ❶ **Benchmark scale.** AgentVista currently focuses on a carefully curated set of 209 tasks, with every instance receiving four hours of expert annotation and verification to ensure quality. We further confirm in Appendix B.2 that the accuracy estimate is stable and low-variance under random sub-sampling, indicating that the reported rankings are robust at this scale. In the

future, we plan to further scale up the benchmark and extend coverage to additional modalities such as video and audio. ❷ **Tool environment.** We focus on four widely used tools (web search, image search, page navigation, and code interpreter), following prior work [Li et al., 2025c, Geng et al., 2025] and the de facto toolsets adopted by current multimodal agents. In the future, we plan to evaluate under more diverse tool environments and alternative agent harnesses.

Conclusion. We introduce AGENTVISTA, a benchmark for evaluating generalist multimodal agents on realistic, ultra-challenging tasks that require long-horizon, interleaved tool use grounded in visual evidence. AGENTVISTA contains 209 carefully curated tasks spanning 25 sub-domains across 7 diverse categories, with strict quality control to ensure vision-centric queries and unique, verifiable answers. Experiments across frontier models show that AGENTVISTA is far from solved: even the best-performing model, GPT-5.6-SOL, reaches only 35.41% overall accuracy. The benchmark also elicits long interaction trajectories, with models such as CLAUDE-SONNET-4.6 and GPT-5.1 averaging more than 12 tool turns per task, indicating substantial complexity beyond short tool chains. Further analysis highlights visual grounding and long-horizon tool use as key bottlenecks for current multimodal agents. We hope AGENTVISTA provides a practical benchmark for tracking progress and motivates the development of stronger multimodal agents that can solve complex, multi-step real-world tasks more reliably.

Acknowledgements

We sincerely thank Prof. Junxian He and Prof. Yi R. (May) Fung for their valuable suggestions and guidance on data construction. We also thank all the anonymous reviewers for their constructive comments. This work was supported by the Hong Kong Research Grants Council Early Career Scheme under Grant No. 26218125.

References

Anthropic. Introducing Claude Opus 4.1: State-of-the-Art High-Performance Multimodal AI. https://www.anthropic.com/news/claude-opus-4-1, 2025a.

Anthropic. Introducing claude 4: Sonnet 4 and opus 4. https://www.anthropic.com/news/claude-4, 2025b.

Anthropic. Introducing claude sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5, 2025c.

Anthropic. Introducing claude opus 4.6. https://www.anthropic.com/news/claude-opus-4-6, 2026a.

Anthropic. Introducing claude sonnet 4.6. https://www.anthropic.com/news/claude-sonnet-4-6, 2026b.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, December 2015.

Tajamul Ashraf, Amal Saqib, Hanan Ghani, Muhra AlMahri, Yuhao Li, Noor Ahsan, Umair Nawaz, Jean Lahoud, Hisham Cholakkal, Mubarak Shah, et al. Agent-x: Evaluating deep multimodal reasoning in vision-centric agentic tasks. *arXiv preprint arXiv:2505.24876*, 2025.

Jiawei Chen, Xintian Shen, Lihao Zheng, Zhenwei Shao, Handong Cui, Chaoqun Du, Li Gong, Feng Gu, Xuefeng Hao, Wei He, Jiabang He, Yi Hu, Bin Huang, Shanshan Li, Qizhen Li, Jing Luo, Zide Liu, Xiaobo Liu, Ning Mao, Lifu Mu, Xuhao Pan, Zhiheng Qu, Chang Ren, Xudong Rao, Haoyi Sun, Qian Wang, Shuai Wang, Zhichao Wang, Wei Wang, Lian Wen, Jiqing Zhan, Hongfu Yang, Sheng Yang, Jiajun Yang, Pengfei Yu, Hongyuan Zhang, Bin Zhang, Chunpeng Zhou, Zheng Zhou, Shucheng Zhou, Shuo Xie, Yun Zhu, Hao Ma, Tao Wei, Pan Zhou, and Wei Chen. Mindwatcher: Toward smarter multimodal tool-integrated reasoning, 2026. URL https://arxiv.org/abs/2512.23412.

Christopher Chou, Lisa Dunlap, Koki Mashita, Krishna Mandal, Trevor Darrell, Ion Stoica, Joseph E Gonzalez, and Wei-Lin Chiang. Visionarena: 230k real world user-vlm conversations with preference labels. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3877–3887, 2025.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, et al. Webwatcher: Breaking new frontier of vision-language deep research agent. *arXiv preprint arXiv:2508.05748*, 2025.

Google DeepMind. Gemini 3 flash: frontier intelligence built for speed. `https://blog.google/products/gemini/gemini-3-flash/`, 2025a.

Google DeepMind. A new era of intelligence with gemini 3. `https://blog.google/products/gemini/gemini-3/`, 2025b.

Google DeepMind. Gemini 3.1 pro. `https://blog.google/products/gemini/gemini-3-1-pro/`, 2026.

Xingang Guo, Utkarsh Tyagi, Advait Gosai, Paula Vergara, Jayeon Park, Ernesto Gabriel Hernández Montoya, Chen Bo Calvin Zhang, Bin Hu, Yunzhong He, Bing Liu, et al. Beyond seeing: Evaluating multimodal llms on tool-enabled image perception, transformation, and reasoning. *arXiv preprint arXiv:2510.12712*, 2025.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, Guohai Xu, and Xing Yu. Deepeyesv2: Toward agentic multimodal model. *arXiv preprint arXiv:2511.05271*, 2025.

Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2019.

Guanyu Jiang, Zhaochen Su, Xiaoye Qu, and Yi R. Fung. Xskill: Continual learning from experience and skills in multimodal agents, 2026. URL `https://arxiv.org/abs/2603.12056`.

Xin Lai, Junyi Li, Wei Li, Tao Liu, Tianjian Li, and Hengshuang Zhao. Mini-o3: Scaling up reasoning patterns and interaction turns for visual search. *arXiv preprint arXiv:2509.07969*, 2025.

Junlong Li, Wenshuo Zhao, Jian Zhao, Weihao Zeng, Haoze Wu, Xiaochen Wang, Rui Ge, Yuxuan Cao, Yuzhen Huang, Wei Liu, et al. The tool decathlon: Benchmarking language agents for diverse, realistic, and long-horizon task execution. *arXiv preprint arXiv:2510.25726*, 2025a.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

Ming Li, Jike Zhong, Shitian Zhao, Haoquan Zhang, Shaoheng Lin, Yuxiang Lai, Wei Chen, Konstantinos Psounis, and Kaipeng Zhang. Tir-bench: A comprehensive benchmark for agentic thinking-with-images reasoning. *arXiv preprint arXiv:2511.01833*, 2025b.

Shilong Li, Xingyuan Bu, Wenjie Wang, Jiaheng Liu, Jun Dong, Haoyang He, Hao Lu, Haozhe Zhang, Chenchen Jing, Zhen Li, Chuanhao Li, Jiayi Tian, Chenchen Zhang, Tianhao Peng, Yancheng He, Jihao Gu, Yuanxing Zhang, Jian Yang, Ge Zhang, Wenhao Huang, Wangchunshu Zhou, Zhaoxiang Zhang, Ruizhe Ding, and Shilei Wen. Mm-browsecomp: A comprehensive benchmark for multimodal browsing agents, 2025c. URL `https://arxiv.org/abs/2508.13186`.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

Yujie Lu, Dongfu Jiang, Wenhu Chen, William Yang Wang, Yejin Choi, and Bill Yuchen Lin. Wildvision: Evaluating vision-language models in the wild with human preferences. *Advances in Neural Information Processing Systems*, 37:48224–48255, 2024.

Zixian Ma, Weikai Huang, Jieyu Zhang, Tanmay Gupta, and Ranjay Krishna. m & m's: A benchmark to evaluate tool-use for m ulti-step m ulti-modal tasks. In *European Conference on Computer Vision*, pages 18–34. Springer, 2024.

OpenAI. Introducing openai o3 and o4-mini. `https://openai.com/index/introducing-o3-and-o4-mini/`, 2025a.

OpenAI. Introducing gpt-5. `https://openai.com/index/introducing-gpt-5/`, 2025b.

OpenAI. Gpt-5.1: A smarter, more conversational chatgpt. `https://openai.com/index/gpt-5-1/`, 2025c.

OpenAI. Introducing gpt-5.2. `https://openai.com/index/introducing-gpt-5-2/`, 2025d.

OpenAI. OpenAI o3 and o4-mini system card. System card, OpenAI, April 2025e. URL `https://cdn.openai.com/papers/o3-o4-mini-system-card.pdf`. Released April 16, 2025.

OpenAI. Introducing gpt-5.4. `https://openai.com/index/introducing-gpt-5-4/`, 2026a.

OpenAI. Previewing gpt-5.6 sol: A next-generation model. `https://openai.com/index/previewing-gpt-5-6-sol/`, 2026b.

OpenAI. Introducing gpt-6.1 sol. `https://openai.com/index/introducing-gpt-6-1-sol/`, 2026c.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023.

Scale AI. Vista: Visual–language understanding leaderboard. `https://scale.com/leaderboard/visual_language_understanding`, 2025.

Barry E Stein. *The new handbook of multisensory processing*. Mit Press, 2012.

Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guanjie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, et al. Openthinkimg: Learning to think with images via visual tool reinforcement learning. *arXiv preprint arXiv:2505.08617*, 2025a.

Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, et al. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. *arXiv preprint arXiv:2506.23918*, 2025b.

Xijia Tao, Yihua Teng, Xinxing Su, Xinyu Fu, Jihao Wu, Chaofan Tao, Ziru Liu, Haoli Bai, Rui Liu, and Lingpeng Kong. Mmsearch-plus: Benchmarking provenance-aware search for multimodal browsing agents. *arXiv preprint arXiv:2508.21475*, 2025.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024a.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 7907–7915, 2025.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems*, 37:113569–113697, 2024b.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13084–13094, 2024.

xAI. Grok-4. `https://x.ai/news/grok-4`, 2025.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094, 2024.

John Yang, Carlos E. Jimenez, Alex L. Zhang, Kilian Lieret, Joyce Yang, Xindi Wu, Ori Press, Niklas Muennighoff, Gabriel Synnaeve, Karthik R. Narasimhan, Diyi Yang, Sida I. Wang, and Ofir Press. Swe-bench multimodal: Do ai systems generalize to visual software domains?, 2024. URL `https://arxiv.org/abs/2410.03859`.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, June 2024a.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024b.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

A AGENTVISTA Details

A.1 Dataset Taxonomy of AGENTVISTA

AGENTVISTA covers seven major categories: (1) **Technology**, which includes hardware troubleshooting, engineering analysis, and system configuration grounded in real photos, screenshots, and diagrams; (2) **Commerce**, which includes product selection, pricing and budget calculation, and finance-related reasoning under practical constraints; (3) **Geography**, which includes route planning, map interpretation, location identification, and spatial calculations; (4) **Entertainment**, which includes sports analytics, media and hobby curation, and game-related reasoning; (5) **Society**, which includes everyday tasks such as health and culinary decisions, home maintenance, manual assembly troubleshooting, and plant care; (6) **Academics**, which includes mathematical computation, scientific identification, and data analysis; and (7) **Culture**, which includes cultural knowledge, history-related understanding, and artifact appraisal grounded in visual evidence.

A.2 Data Sources

All AGENTVISTA instances are grounded in real images and real user needs. Across all sources, we apply a unified set of criteria. We retain only images with sufficient visual detail to support non-trivial reasoning, and we exclude cases where the solution can be obtained by directly searching the query text or by retrieving the same image and question from the public web. We curate candidates from three channels.

Public user-submitted arenas. We collect image-based user submissions from public vision-language model arenas, including VisionArena and WildVision [Chou et al., 2025, Lu et al., 2024]. This source provides 284.4K images with diverse real-world scenes. We first apply an automated filter using CLAUDE-OPUS-4 to remove images with limited visual information and cases that do not fit agentic problem settings. The filter also proposes a candidate task query that reflects the plausible action space. The prompt is shown in Appendix B.5.1. Human annotators then select high-quality candidates for downstream curation.

Annotator-captured real-life scenarios. We also include tasks collected by annotators from real daily situations, together with the original photos or screenshots that motivated the request. This channel naturally captures practical constraints, such as cluttered scenes, partial evidence, and ambiguous context, which are common in real deployments. We treat these instances as first-party user needs and keep their intent while ensuring the final task remains self-contained.

Private community forums. We also curate candidates from community help-seeking forums. We collect posts that include visually informative images and reflect realistic user goals. Since these posts often contain lengthy discussions and personal details, we rewrite each case into a standalone task while preserving the original intent and removing identifying information. We apply stricter screening to ensure clarity and consistency with our benchmark standards.

B Experimental Details

B.1 Tool Definition

AGENTVISTA is evaluated in a controlled tool environment with a compact set of commonly used tools for multimodal agent workflows. Models can invoke these tools appropriately within the `<tool_call>...</tool_call>` block during interaction. In detail, our tools are defined as follows.

Tool: Web Search

Description: Search the web for information online. Use when you need to find information, facts, or current events. Returns web search results with titles, URLs, and text snippets. You only have limited search times, so please use it wisely.

Parameters:
- **query**: Search query string (required)

- **max_results**: Maximum number of results to return (default: 10)

Tool: Image Search

Description: Search for related images using text query or reverse image search.
- For text-to-image search: specify search_type="text" and provide a query.
- For reverse image search: specify search_type="reverse" and provide an image_url.
Returns images with URLs and descriptions.

Parameters:
- **search_type**: Type of search ('text' or 'reverse', default: 'text')
- **query**: Search query string (required for 'text' mode)
- **image_url**: Image filename, local reference, or URL (required for 'reverse' mode)
- **max_results**: Maximum number of image results to return (default: 10)

Tool: Visit

Description: Visit a webpage and extract its main content. Use when you have a specific URL to visit (often after getting a URL from web search or image search). Extracts and returns the main textual content of the webpage.

Parameters:
- **url**: Full URL of the webpage to visit (must start with 'http://' or 'https://')
- **goal**: What information you want to find on this page (helps focus the extraction)

Tool: Code Interpreter

Description: Executes Python code in a stateful Jupyter kernel and return results.

Capabilities:
- Use PIL, NumPy, or OpenCV to process or analyze images to improve understanding.
- Perform complex mathematical calculations or data manipulation.
- Code execution is persistent: variables and functions are stored and reusable in subsequent calls.

Image Variables (Pre-loaded as PIL Image objects):
- **original_image / original_image_N (N=1,2,3,...):** The original input images are already loaded as PIL Image objects.
- **tool_image_N (N=1,2,3,...):** Images generated by your previous code calls.
- **observation_N (N=1,2,3,...):** Images generated by zoom tool operations.

Visualization & Output:
- To view generated images, return the object, use `plt.show()`, or save it (e.g., `image.save('out.png')`).
- DO NOT use `image.show()`.
- Displayed images will be available as `"tool_image_N"` in the next turn.

Pre-installed packages:
- PIL, NumPy, OpenCV, Matplotlib, SciPy, Scikit-learn, Scikit-image, Pandas, SymPy

Parameters:
- **code**: Python code to execute in the Jupyter kernel (required)

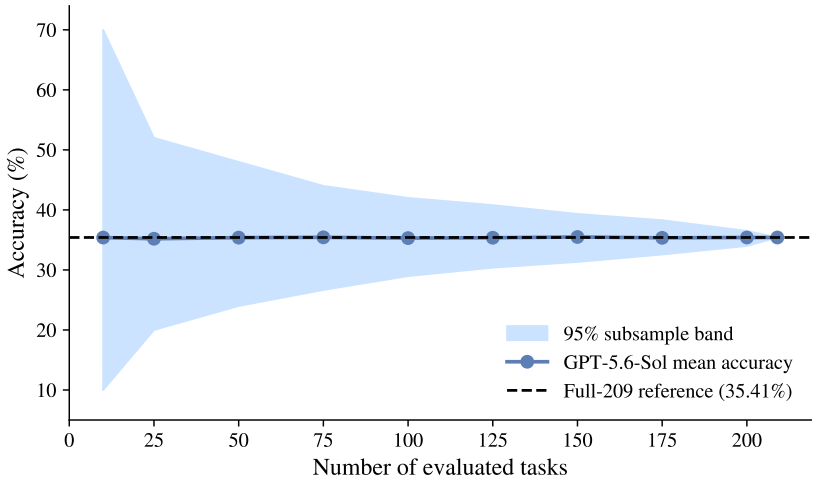


Figure 8: Sub-sampling analysis on AGENTVISTA. For each k, we draw 2,000 random subsets from the 209-task pool and compute GPT-5.6-SOL's accuracy on each. The estimate is stable across all k, with the mean equal to the full-set value and the 95% subsample interval shrinking at a $1/\sqrt{k}$ rate.

Table 5: Results of representative open-source models on AGENTVISTA by category. Domain abbreviations: **COMM.** (Commerce), **GEOG.** (Geography), **ENT.** (Entertainment), **TECH.** (Technology), **SOC.** (Society), **ACAD.** (Academics), and **CULT.** (Culture). The best-performing model in each category is **bold**, and the second best is underlined. All values are accuracies in %.

MODEL	COMM.	GEOG.	ENT.	TECH.	SOC.	ACAD.	CULT.	OVERALL
QWEN3-VL-235B	7.14	10.26	**10.26**	**26.47**	16.00	**20.00**	13.33	**13.88**
DEEPEYES-V2-7B	**9.52**	10.26	5.13	14.71	24.00	6.67	**20.00**	11.96
WEBWATCHER-32B	0.00	**12.82**	2.56	23.53	**28.00**	**20.00**	0.00	11.48

B.2 Sub-sampling analysis

We have reported the results of different models in Table 3. A natural question is whether 209 tasks are sufficient to yield stable rankings. In this section, we analyze how the accuracy estimate varies with subset size by sub-sampling the 209-task pool.

Experimental setup. We evaluate the leader GPT-5.6-SOL on random subsets of varying size k drawn from the 209-task pool. For each $k \in \{10, 25, 50, 75, 100, 125, 150, 175, 200, 209\}$, we draw R=2,000 subsets without replacement, compute strict accuracy on each subset, and record the mean together with the 2.5 and 97.5 percentiles. The results are shown in Figure 8.

Key findings. Three observations stand out. First, the resample mean equals the full-set accuracy of 35.41% at every k, including the smallest setting k=10, so the reported number is not driven by a few unusually easy or hard tasks, and any random selection of 209 tasks from a similar pool would yield the same expected accuracy. Second, the 95% subsample interval shrinks predictably from $\pm$30 points at k=10 to $\pm$6.5 points at k=100 and $\pm$1.2 points at k=200, so the variance of the accuracy estimate at our scale is small. Third, this half-width is well below the multi-point gaps separating frontier models in Table 3, so the reported rankings are robust to which specific 209 tasks were curated. One exception is the ordering between the two newest models, GPT-5.6-SOL and GPT-6.1-SOL, which are only 0.96 points apart, within decoding and judging noise. In short, n=209 already yields a low-variance, stable estimate of model accuracy on AGENTVISTA.

B.3 Analysis of open-source model results.

Table 5 reports results for three representative open-source multimodal models. In particular, DEEPEYES-V2-7B [Hong et al., 2025] and WEBWATCHER-32B [Geng et al., 2025] are tool-using open-source agents that can interact with external tools to support multi-step problem solving, while

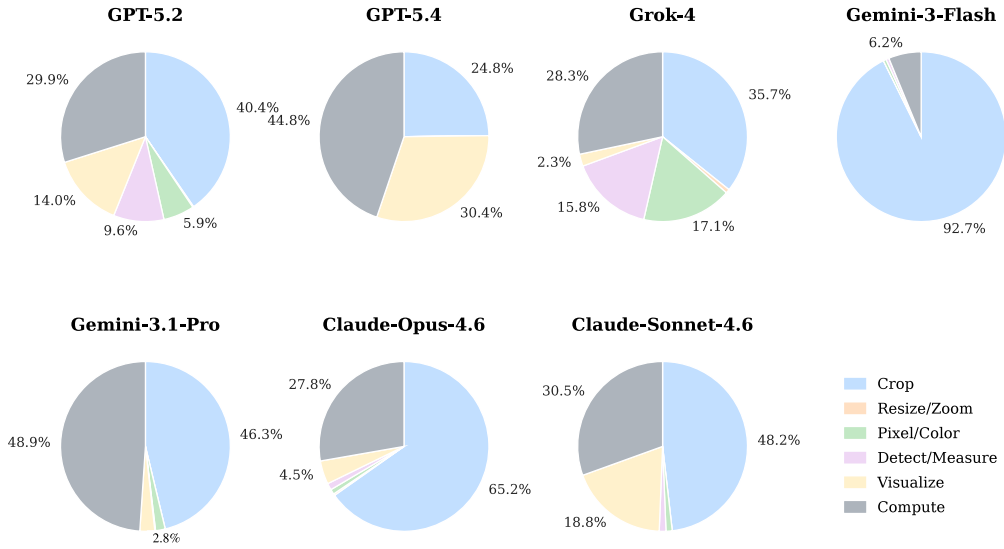


Figure 9: Distribution of code-interpreter operations across the selected models. Each call is parsed and assigned to six main operation types: *Crop*, *Resize/Zoom*, *Pixel/Color* sampling, *Detect/Measure*, *Visualize*, and *Compute*.

QWEN3-VL-235B serves as a strong open-source multimodal backbone. Overall, these open-source baselines remain far from solving AGENTVISTA, i.e., their overall accuracy ranges from 11.48% to 13.88%, substantially lower than the best-performing closed-source model GPT-5.6-SOL at 35.41%. This gap further reflects the ultra-challenging nature of AGENTVISTA and highlights the large room for improving open-source multimodal agents.

B.4 Code-interpreter operation breakdown

In this section, we further break down code interpreter calls by operation type. We pattern-match the Python source of every `code_interpreter` call against a fixed taxonomy. As shown in Figure 9, for most models, *crop* is the most frequent operation, indicating that many trajectories depend on localized visual grounding before proceeding to retrieval or computation. The results suggest that these models more often perform image-centric operations during problem solving, such as zooming in, cropping, resizing, measuring regions, and carrying out structured extraction or calculations. Second, GROK-4 stands out as the most analytically diverse, with substantial pixel and color sampling and explicit detection or measurement, indicating a stronger reliance on quantitative image inspection than other models. Third, GPT-5.4 frequently invokes the code interpreter, but most calls focus on visualization and pure computation rather than image manipulation. Across all models, structured *Detect/Measure* operations remain comparatively rare, suggesting that current agents fall back to direct visual reading rather than programmatic measurement.

B.5 Prompts

B.5.1 Prompts for Data Construction

Agentic Benchmark Task Filtering Criteria

Objective: You are tasked with evaluating whether images are suitable for designing agent tasks with VERIFIABLE, UNIQUE ANSWERS, which require MULTIPLE SEARCHES and COMPLEX MULTI-HOP REASONING.

Core Requirements (All must be satisfied):

- **Complex Visual Content:** Rich, detailed visual elements (e.g., product catalogs, maps, menus, data tables, schedules) that support real-world task formulation.

- **Multiple Searches Required:** At least 2-3 distinct search operations are needed to gather diverse external information for solving the task.
- **Complex Multi-hop Reasoning:** The task must involve at least 4-5 reasoning steps that build on one another to arrive at the final solution.
- **Tool Synergy:** Tasks must leverage [Multiple Search/Browser operations + Complex Code execution] for an optimal solution.
- **Real-world Scenarios:** Tasks should be based on real-world scenarios, such as travel planning, shopping decisions, event scheduling, or data analysis.
- **Verifiable Unique Answer:** The task must lead to a verifiable and unique answer, which can be one of the following:
  - A number (e.g., “42", “3.14", “125", “15km")
  - A short string (e.g., “Brittany", “October 2, 2025", “Paris")
  - A date/time (e.g., “2025-10-02", “14:30", “March 15")
  - A location name (e.g., “Central Park", “Tokyo Station")
  - A product/item name (e.g., “iPhone 15", “Margherita Pizza")

Must Reject:

- **OCR Tasks:** Pure text recognition, translation, or transcription.
- **Simple Images:** Anime characters, simple landscapes, basic objects, or cartoons.
- **Academic Tasks:** Tasks involving papers, formulas, or professional charts.
- **Direct Q&A:** Questions that can be answered directly without the need for external tools.
- **Too Simple:**
  - Less than 3 reasoning steps.
  - Single-hop reasoning (e.g., “Find the price of X").
  - Simple calculations that lack complex logic.
- **Non-verifiable Answers:**
  - Plans or itineraries (e.g., “Design a travel plan").
  - Reports or summaries (e.g., “Write a shopping report").
  - Recommendations or suggestions (e.g., “Recommend 3 hotels").
  - Long explanations or descriptions.
  - Multiple answers or lists (unless asking for a count).
  - Subjective opinions.

B.5.2 The Prompt for Evaluation

Multimodal Agent Prompt

You are a visual reasoning agent. Your goal is to answer questions about images.

AVAILABLE TOOLS:

- **web_search:** Search the web for information, facts, or current events.
- **image_search:** Search for related images using text query or reverse image search.
- **visit:** Visit a webpage and extract its main content.
- **code_interpreter:** Execute Python code for image processing, analysis, and calculations.

INSTRUCTIONS:

1. **Analyze:** Carefully observe the image and the user’s question.
2. **Think:** Explain your step-by-step reasoning process.
3. **Use Tools:** Call the appropriate tool to gather information and help answer the question.
4. **Iterate as needed:** Continue reasoning and using tools in next turns until you are confident in your findings.

5. **Answer:** Once confident, provide the final answer inside `<answer>...</answer>` tags!

IMPORTANT:
- Always explain your detailed reasoning process before using any tool.
- You can ONLY call one tool at a time! Do not call multiple tools in one turn!
- You MUST provide your final answer using complete `<answer>...</answer>` tags!

C Error Type Definitions

In Section 4.3, we report the error distributions of representative models on AGENTVISTA. Here we define the error types used in our taxonomy.

Tool execution failure. This category captures cases where the agent follows a plan, but fails due to issues in tool interaction. Typical examples include empty tool outputs, invalid requests, and failures to open or parse retrieved content. These errors suggest that robust tool use and self-checking are important for completing long-horizon workflows.

Visual misidentification. This category includes errors caused by incorrect visual understanding, such as reading the wrong text on a label, confusing similar components, missing a small indicator, or miscounting objects. Because visual evidence often determines what to search for and how to apply constraints, a single perception mistake can cause later steps to follow an incorrect direction.

Knowledge hallucination. This category refers to cases where the agent outputs facts that are not supported by the provided images or retrieved sources. Common patterns include inventing details that look plausible, relying on generic rules of thumb, or asserting standards that do not match the evidence in the current instance. These failures indicate insufficient grounding in the multimodal context.

Calculation error. This category covers mistakes in arithmetic or multi-step aggregation, such as wrong unit conversions, incorrect date computations, or errors when combining multiple retrieved values. These cases often arise after several steps, when the agent must keep intermediate numbers consistent while continuing to use tools.

Instruction misinterpretation. This category includes failures to follow the user request or constraints, such as ignoring a time window, missing a required format, applying the wrong condition, or answering a related but different question. Even when perception and retrieval are correct, misunderstanding the intent can still lead to an incorrect final answer.

Others. This category groups remaining failures that do not fit the above types or that involve multiple types without a clear primary cause. Examples include incomplete final answers, premature termination, inconsistent outputs across steps, or cases where the model produces an answer that cannot be checked against the required format. We use this bucket to keep the taxonomy simple while still accounting for long-tail error patterns.

D Broader Impacts

This work introduces AGENTVISTA, a benchmark for evaluating generalist multimodal agents on realistic, ultra-challenging tasks that require long-horizon tool use grounded in real visual inputs. By using concise, verifiable answers and a controlled tool environment, AGENTVISTA enables reproducible comparisons and helps identify key bottlenecks in visual grounding, constraint tracking, and tool reliability. Improved multimodal agents could benefit practical applications such as shopping assistance, travel planning, and troubleshooting from user photos, where agents must combine visual evidence with online information and computation. At the same time, stronger agents may increase risks of privacy leakage from user-provided images and overconfident but incorrect outputs in real deployments. We mitigate these concerns by filtering and rewriting tasks to avoid personal identifiers

when applicable, and by emphasizing short answers that encourage checkable evaluation rather than persuasive free-form text.

Benchmark construction can also reflect biases from source data and annotator decisions, which may affect coverage across domains and scenarios. We hope AGENTVISTA supports future work on more robust and responsible multimodal agents by providing a shared evaluation target for realistic, long-horizon tool use.

E Case Study

In this section, we present representative trajectories to illustrate both successful and failed behaviors on AGENTVISTA. We first show a good-case example that demonstrates effective long-horizon, interleaved tool use. We then provide one bad-case example for each error type, highlighting how different failure modes arise and how they derail the overall workflow.

E.1 Good Case Examples

Traj #1: Sneaker Authentication. This task involved verifying the authenticity of luxury sneakers based on visual evidence. Through a sequence of seven tool invocations, the model conducted a systematic examination of specific features. It utilized Image Search to contrast tongue and size tags with authentic references, identifying an anomalous "A8513" sticker. Subsequent validation via Web Search confirmed this as a counterfeit indicator, leading to the correct classification.

Traj #2: Strongest German Beer Analysis. Identifying the strongest beer required distinguishing specific brands within a cluttered image. The model synergized the Code Interpreter for visual refinement with Web Search for factual retrieval. This approach enabled the precise filtering of lower-alcohol options, resulting in the accurate identification of a tie between Steam Brew German Red and Perlenbacher Strong.

Traj #1: Sneaker Authentication (GEMINI-3-PRO; success; 7 tool calls)

Task. Acting as a luxury sneaker authenticator, search for authentic craftsmanship images for comparison. Decide whether the shoes are **Real** or **Fake**, and give at least two visual reasons.

Input images.



Ground truth. Counterfeit. Key cues include Widow's Peak Error, Poor Tongue Font & Stitching, Misaligned 3M Stripes.

Tool trajectory (condensed).
Step 1. *web search*
Query: `Dior B30 black sneaker authentic vs fake guide`
Evidence. Retrieved legit-check guides that highlight tongue tag font, stitching, and interior label formats as key checks.

Step 2. *image search*
Query: `authentic Dior B30 black sneaker tongue tag`
Evidence. Collected reference images for tongue tag layout and font.



Step 3. *image search*
Query: `authentic Dior B30 sneaker inside size tag label`
Evidence. Collected reference images for inside label formats and placement.



Step 4. *web search*
Query: `Dior B30 authentic inner size code format`
Evidence. Guides report authentic style identifiers follow a structured style-ID pattern, rather than generic sticker-like codes.

Step 5. *image search*
Query: `fake Dior B30 size sticker A8513`
Evidence. Collected examples where generic interior stickers are associated with replicas.



Final decision. Fake.
Reasons.
- The interior sticker code format is inconsistent with authentic labeling patterns shown in retrieved references.
- The tongue tag font and stitching do not match the authentic reference images. ✓

Traj #2: Strongest German Beer Analysis (GEMINI-3-PRO; success; 7 tool calls)

Task. From the beers shown in the photo, consider only German-brewed beers with ABV $> 5\%$. Which **brand** has the highest **total alcohol per can**, accounting for both ABV and can volume?

Input image.



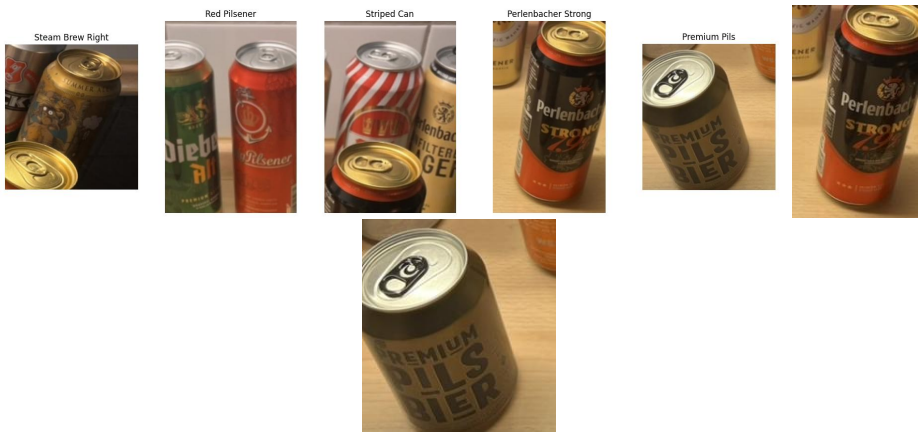
Ground truth. Steam Brew German Red and **Perlenbacher Strong** (tie).

Tool trajectory (condensed).
Step 1. *code interpreter*
Evidence. Zoomed and cropped the photo to read brand names and the ABV/volume information on the cans.



Step 2. *web search*
Query: `Perlenbacher Strong ABV  Steam Brew German Red ABV`
Evidence. Retrieved product pages confirming both beers are **7.9% ABV**.

Step 3. *code interpreter*
Evidence. Cropped individual cans to verify they are German-brewed and to confirm can capacity (standard tall can).



Step 4. *web search*
Query: `Steam Brew German Red volume 500ml`
Evidence. Confirmed the can size used for calculation is **500 ml**.

Step 5. *image search*
Evidence. Attempted to identify other cans/variants; no higher-ABV German-brewed option was found among the eligible beers.

Step 6. *web search*
Query: `Steam Brew German Red ABV exact  Perlenbacher Strong ABV`
Evidence. Cross-checked ABV values and resolved remaining ambiguity.

Final decision. Tie: Steam Brew German Red and Perlenbacher Strong.
Reason. Both are German-brewed, exceed 5% ABV, and have the same alcohol per can: $0.079 \times 500\ \text{ml} = 39.5\ \text{ml}$. ✓

E.2 Bad Case Examples

Traj #3: Karst Jigsaw Puzzle. ***Tool execution failure*****.** **Task.** Reconstruct a 6×6 jigsaw puzzle from an input image and locate the missing piece position. **Failure.** The model attempted to segment puzzle pieces with code-based image processing, but the segmentation failed and extracted only 24 segments instead of the expected 35. Without a complete set of pieces, the model could not form a valid grid and the reconstruction became infeasible. **Classification Rationale.** The core issue is a breakdown in tool-based image processing, which blocks the workflow even though the high-level plan is reasonable.

Traj #4: Authors United Window Display. ***Visual misidentification*****.** **Task.** Identify the author shown in a window display from the provided image. **Failure.** The visible author is Donna Tartt, but the model failed to identify her. Although it performed cropping, it still did not extract the correct visual cue and produced an incorrect identification. **Classification Rationale.** The decisive evidence is in the image, and the failure comes from incorrect visual recognition rather than retrieval or reasoning.

Traj #5: Target Arena Identification. ***Visual misidentification*****.** **Task.** Identify the correct university basketball facility shown in the image. **Failure.** The model misread an unclear floor logo and anchored on the wrong university, then reinforced the mistake using generic features such as roof trusses. It concluded the venue was St. Thomas AARC, while the correct answer is UNC. **Classification Rationale.** The initial mistake is a wrong visual anchor, and later steps follow that incorrect anchor.

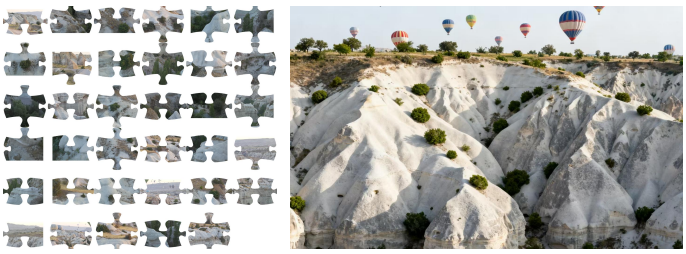
Traj #6: Pilea Root Diagnosis. ***Knowledge hallucination*****.** **Task.** Diagnose the hard mass on Pilea roots from the image. **Failure.** The correct interpretation is calloused residue from root rot, but the model claimed it was a "nursery plug" or fungal material and described visual properties that are not supported by the image. The final diagnosis followed a made-up interpretation aligned with retrieval results rather than the provided evidence. **Classification Rationale.** The model introduces unsupported facts and forces the image to fit a preconceived explanation.

Traj #7: Studio Swing Prop Design. ***Instruction misinterpretation*****.** **Task.** Design a stationary photo prop that visually looks like a suspended swing. **Failure.** The model proposed a design where the seat is visibly supported by a horizontal bar, which removes the hanging illusion and violates the core constraint of the request. **Classification Rationale.** The model fails to follow the key constraint and answers a different problem than the one asked.

Traj #3: Karst Jigsaw Puzzle (GEMINI-3-PRO; failed; 3 tool calls)

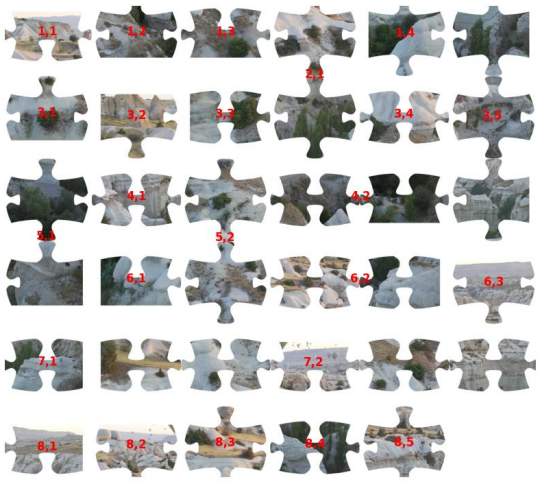
Task. The first image shows a 6×6 jigsaw puzzle (one piece missing) made from a karst-landform photo. The second image is a local reference landscape photo. Reconstruct the puzzle and report the coordinate of the missing piece. Coordinates are defined on the reconstructed grid: top-left is $(1, 1)$, top-right is $(1, 6)$, and bottom-right is $(6, 6)$.

Input images.



Ground truth. $(3, 1)$

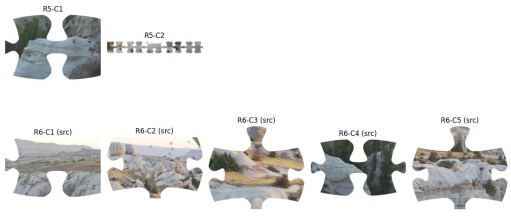
Tool trajectory (condensed).
Step 1. *code interpreter*
Goal. Segment the puzzle into individual pieces and assign each piece to a grid cell.
Outcome. The segmentation merges adjacent pieces and fails to recover all pieces.



Step 2. *code interpreter*
Goal. Improve segmentation with morphological operations and contour filtering.
Outcome. Still under-segments the puzzle and extracts only **24** regions, instead of the expected **35** pieces (or 36 slots).



Step 3. *code interpreter*
Goal. Crop and visualize the extracted pieces for downstream matching.
Outcome. Visualization confirms missing segments; reconstruction cannot proceed.



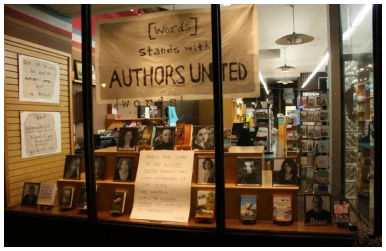
Failure point. Step 2 (segmentation). The model cannot reliably separate touching pieces, so it fails to obtain a complete set of puzzle pieces. Without correct piece extraction, subsequent grid reconstruction and missing-cell identification are not feasible.

Final outcome. The model fails to reconstruct the 6×6 layout and cannot determine the missing coordinate. ✗

Traj #4: Authors United Window Display (GEMINI-3-PRO; failed; 13 tool calls)

Task. This photo shows a bookstore window display supporting "Authors United," featuring photos of authors who have appeared at the bookstore. Among the authors whose photos are clearly visible in the display, identify the author whose work stayed on *The New York Times* Bestseller List for the most total weeks during the years when "Authors United" was most active. Report the author, the work, and the total weeks.

Input image.



Ground truth. Donna Tartt; *The Goldfinch*; 39 weeks.

Tool trajectory (condensed).
Step 1. *web search*
Goal. Identify the bookstore and locate a source describing the window display.
Outcome. Finds a relevant local article about the display.
Step 2. *visit*
Goal. Read the article and recover the set of authors shown in the display.
Outcome. The page provides context but does not yield a complete, reliable list of the visible authors.
Step 3. *code interpreter*
Goal. Extract author names from the image via OCR.
Outcome. OCR is noisy and incomplete; text cues are not recovered reliably.
Step 4. *code interpreter*
Goal. Crop the window display into smaller regions to identify individual authors.
Outcome. Produces multiple crops, but faces and cover details remain ambiguous at this resolution.



Steps 5–9. *image search / web search / visit*
Goal. Identify cropped portraits and posters (e.g., "READ" poster) via reverse search and related pages.
Outcome. Reverse search does not return actionable matches; the model cannot reliably map crops to specific authors.

Steps 10–13. *web search*
Goal. Estimate the correct answer by comparing bestseller-list durations among guessed candidates.
Outcome. The model anchors on the wrong author set and proceeds with an incorrect comparison.

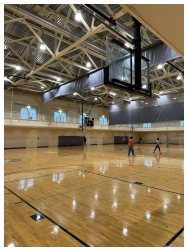
Failure point. Step 4 (visual identification). Even after cropping, the model fails to correctly identify the clearly visible author (Donna Tartt) from the display. This incorrect visual grounding leads to downstream searches and bestseller comparisons over the wrong candidate set, culminating in an incorrect final answer.

Final answer. Anthony Doerr; *All the Light We Cannot See*; ∼84 weeks. ✗

Traj #5: Target Arena Identification (GEMINI-3-PRO; failed; 11 tool calls)

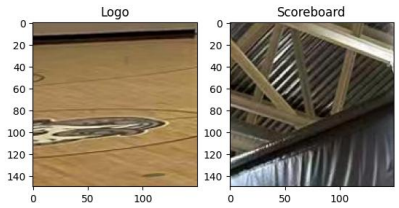
Task. A colleague referenced this venue only as "Target Arena A." Identify exactly which university facility is shown so the correct team metadata (e.g., seating capacity, fan distribution) can be linked in a database.

Input image.



Ground truth. UNC

Tool trajectory (condensed).
Step 1. *code interpreter*
Goal. Inspect the full scene and locate the most informative identifiers (court logo, scoreboard, signage).
Outcome. The model focuses on the floor logo and plans to use it for identification.
Step 2. *code interpreter*
Goal. Crop the court logo and scoreboard region for downstream search.
Outcome. Produces a small, low-detail logo crop (insufficient for reliable matching).



Step 3. *image search (reverse)*
Goal. Identify the venue or team via reverse search on the cropped logo.
Outcome. No matches returned.
Step 4. *code interpreter*
Goal. Re-crop the logo with more context.
Outcome. The crop remains ambiguous and visually distorted.



Step 5. *image search (reverse)*
Goal. Retry reverse search with the updated crop.
Outcome. Still no matches.

Step 6. *web search*
Goal. Use structural cues (exposed truss ceiling, running track/balcony) to narrow candidates.
Outcome. Returns generic lists; does not provide a grounded identification.
Step 7. *code interpreter*
Goal. Enhance contrast/sharpness of the logo crop to clarify the mascot shape.
Outcome. Enhancement does not resolve the mascot identity.



Step 8. *image search (reverse)*
Goal. Reverse search the enhanced crop.
Outcome. No matches.
Step 9. *image search (text)*
Goal. Pivot to a guessed candidate (University of St. Thomas AARC) based on weak visual similarity.
Outcome. Retrieves reference images for comparison.
Step 10. *image search (text)*
Goal. Expand references (Schoenecker Arena / AARC interiors) and match architecture.
Outcome. The model over-weights generic ceiling/track similarities.
Step 11. *image search (text)*
Goal. Final confirmation via additional reference images.
Outcome. Commits to the incorrect university.

Failure point. Step 9 (incorrect anchoring). After failing to identify the floor logo, the model switches to architecture-based matching and prematurely anchors on St. Thomas. The remaining steps reinforce this guess using generic similarities (trusses, ducts, balcony/track) rather than a definitive visual identifier from the query image, leading to an incorrect final answer.

Final answer. University of St. Thomas (Minnesota) ✗

Traj #6: Pilea Root Diagnosis (GEMINI-3-PRO; failed; 6 tool calls)

Task. A Pilea was overwatered two months ago and later stabilized. During repotting, a large white mass is found at the root base; it is hard and crusty (does not burst when poked). Based on the images and history, identify the substance and decide whether pesticides are needed.

Input images.



Ground truth. No pesticide needed. The mass is calloused / crustified root-rot residue from the prior overwatering incident.

Tool trajectory (condensed).
Step 1. *web search*
Goal. Identify common causes of a hard white mass near Pilea roots.
Outcome. Results skew toward unrelated "white spots" / mineral topics (mostly leaf-focused).
Step 2. *web search*
Goal. Broaden to generic houseplant root "white lump" explanations.
Outcome. Mixed hypotheses (mineral buildup, mold/fungus) without matching the specific context.

Step 3. *image search*
Goal. Look up nursery-plug examples as an alternative explanation.
Outcome. Retrieves plug images that encourage an external "plug/mesh" hypothesis.
Step 4. *web search*
Goal. Explore other possibilities (e.g., slime mold / residue) from search cues.
Outcome. Finds generic statements that crusty residue can occur, but without strong alignment to an underground root-base mass.
Step 5. *web search*
Goal. Re-search with Pilea-specific phrasing to confirm the plug/mesh theory.
Outcome. Surfacing discussions about "root mesh / death plugs" further reinforces the plug narrative.
Step 6. *web search*
Goal. Finalize between plug/mesh vs. plant-tissue explanation.
Outcome. Commits to "nursery plug + dried fungus" as the main explanation.

Failure point. Step 3 (incorrect hypothesis shift). After early searches returned noisy, non-root-specific results, the model pivots to an external "nursery plug / mesh" explanation based on loosely similar reference images. This guess then becomes the anchor and is reinforced by subsequent searches, while the overwatering history and root-location cue (consistent with healed root damage) are under-weighted.

Final answer. A nursery plug ("death plug") encrusted/colonized by dried saprophytic fungus; no pesticides. ✗

Traj #7: Swing Prop Construction (GEMINI-3-PRO; direct response)

Task. Build a swing prop for a children's photography studio that *looks* like a rope-hung swing but is completely stationary and safe. The seat must appear connected to the top beam (no "floating" look). Only basic woodworking tools are available.

Input image.



Ground truth. Bolt the seat to the frame using vertical wooden posts, then wrap thick hemp rope around the posts to hide the wood and create the hanging illusion.

Model response (condensed).
Step 1 (direct).

- **Interpretation.** The model treats the prop as a bench-like structure supported by a visible horizontal stretcher bar, rather than creating a concealed rigid support that still looks rope-hung.
- **Proposed design.** Build a rigid A-frame, add a horizontal stretcher across the legs, mount the seat on the stretcher, and add decorative (non-load-bearing) ropes from top beam to seat to mimic a swing.

Failure point. Constraint miss (visual illusion). The proposed stretcher-bar support remains visually apparent and undermines the requirement that the seat should look convincingly hung from the top beam. It does not use hidden vertical supports wrapped in rope, which is the key trick in the ground-truth solution.

Final answer. Build a stationary bench supported by a horizontal bar, then add taut decorative ropes to imitate a swing. ✗

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The Abstract and Section 1 clearly state the contributions and scope of AGENTVISTA, including the 209 carefully curated tasks across 25 sub-domains and 7 categories, the controlled tool environment, and the key empirical findings (e.g., the best model GPT-5.6-SOL reaches only 35.4% overall accuracy).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in the *Limitations* paragraph of Section 6 (Discussion), covering benchmark scale and tool-environment scope, together with our plans to expand both in future work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This work presents an empirical benchmark and contains no theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 3.1 specifies the evaluation protocol, including model list, sampling temperature, the 30-turn tool budget, and the GPT-5.6-LUNA judge. Appendix B.1 provides full tool definitions, and Appendix B.5.1 releases the prompts used for data construction and evaluation. We will additionally release the benchmark and evaluation harness in the supplementary materials.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the benchmark, evaluation harness, prompts, and instructions for reproducing the main results in the supplementary materials. The project page is `https://agentvista-bench.github.io/`, and the evaluation code is available at `https://github.com/hkust-nlp/AgentVista`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: AGENTVISTA is an evaluation benchmark with no training. The full evaluation setting (model list, temperature, tool budget, judge model) is given in Section 3.1, with tool definitions and prompts in Appendix B.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: We report a sub-sampling analysis on the 209-task pool in Appendix B.2, drawing R=2,000 random subsets at each size k and computing accuracy on each. The 95% subsample interval shrinks from $\pm$30 points at k=10 to $\pm$1.2 points at k=200, which is well below the multi-point gaps separating frontier models in Table 3, indicating that the reported rankings are statistically robust at our scale, except between GPT-5.6-SOL and GPT-6.1-SOL, whose 0.96-point gap lies within decoding and judging noise.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All closed-source model evaluations are conducted through their respective APIs, with average tool-calling trajectories in the range of 1.89–17.14 turns per task as reported in Table 3. The full evaluation across 18 closed-source models on 209 tasks, together with the tool-ablation, test-time-scaling, and error-labeling experiments, consumes on the order of 10^8 input tokens and 10^7 output tokens, with total API expenditure of USD $4,500. Open-source baselines are served on 8 A100 GPUs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that this work conforms to its requirements, including data sourcing with appropriate consent, anonymization of personally identifiable information, and fair compensation of annotators.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Both positive and negative societal impacts are discussed in Appendix D (Broader Impacts), including potential applications in shopping, travel, and troubleshooting assistance, as well as risks around privacy leakage, overconfident outputs, and dataset bias, together with mitigation strategies.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: AGENTVISTA contains no model weights and consists of curated, anonymized images and text. Personally identifiable information is filtered and rewritten during data construction, as described in Section 2.2.1 and Appendix A.2. We will release the benchmark under a research-only license that prohibits commercial use and use as model training data, accompanied by usage guidelines.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All third-party assets (e.g., VisionArena and WildVision for source images; the OpenAI, Anthropic, Google, xAI, and open-source model APIs and weights for evaluation) are credited with citations to their original papers and used under their default research-permissive licenses and terms of service.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: AGENTVISTA is released with documentation describing its taxonomy (Appendix A.1), data sources (Appendix A.2), tool environment (Appendix B.1), and intended research-only use, included alongside the benchmark in the supplementary materials.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

Justification: The annotation pipeline is described in Section 2.2.1 and the data construction stages (Section 2.2). Annotators are recruited as expert contributors, undergo training on the project taxonomy and tool environment, and spend on average four hours per task on construction and verification.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The annotation work involves no sensitive personal data and poses minimal risk to contributors.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: All LLM usage in our pipeline is documented in the paper: CLAUDE-OPUS-4 is used as the Stage 1 image filter (Section 2.2.2 and Appendix A.2); GEMINI-3-FLASH is used for execution filtering and for generating error-type labels (Section 2.2.2 and Section 4.3); and GPT-5.6-LUNA is used as the evaluation judge (Section 3.1).

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.