AGENTVQA: A UNIFIED BENCHMARK FOR AGENTIC VISUAL UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) can perform a broad range of tasks across diverse settings, yet their performance in agentic contexts remains poorly understood. Although online evaluations in simulators and real-world environments are essential for measuring interactive agent performance, they are costly and challenging to scale. Complementary offline VQA benchmarks can enable faster and more reliable comparisons across agentic domains, yet existing benchmarks remain limited and fragmented. To address this gap, we introduce AgentVQA, an offline benchmark for evaluating VLMs on realistic agentic scenarios across five agent domains: Web Agents, Robotics, Egocentric Videos, Games, and Spatial Planning. We build AgentVQA by manually selecting challenging, real-world agent examples and adapting them for reliable offline evaluation. The resulting benchmark preserves realistic scenarios from interactive environments across diverse domains, and its model rankings track those from corresponding online evaluations. Across all models evaluated, the best achieves only 58.5% accuracy. Our cross-domain analysis shows that errors are driven primarily by failures to ground and maintain accurate representations of the current and evolving visual state, including localizing relevant UI elements, interpreting motion, tracking state changes over time, and inferring 3D spatial relationships, rather than by task misinterpretation or downstream logical reasoning.



Figure 1: **The AgentVQA Benchmark.** AgentVQA unifies 23 challenging datasets across five domains: Web Agents, Egocentric Videos, Robotics, Games, and Spatial Planning, into a standard MCQ format. These tasks are diverse, with some using action histories to test episodic memory and others using videos to evaluate temporal understanding.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Vision-Language Models (VLMs) are quickly becoming the decision-making core for agentic systems spanning robotics, wearable assistants, web navigation, or gameplay. By prompting or fine-tuning, VLMs often outperform their domain-specific counterparts (Koh et al., 2024; Black et al., 2024; Zhou et al., 2025; Zhang et al., 2025; Sarch et al., 2025) and serve as the backbone for the top-performing action models (Black et al., 2026; 2025; Ji et al., 2025).

In domains such as mathematics and coding, standardized benchmarks have accelerated progress by systematically tracking model capabilities and reliably exposing performance gaps (Jimenez et al., 2023; Balunović et al., 2025). For agents, online evaluations in real or simulated environments remain the gold standard for assessing agentic performance (Yao et al., 2024), but due to computational cost and reproducibility, they present significant practical challenges for rapid, large-scale model comparison (Henderson et al., 2018; Dasari et al., 2022).

As a result, VLMs' agentic capabilities lack comparable evaluation frameworks. State-of-the-art models are often assessed on general-purpose benchmarks like MMMU (Yue et al., 2025), MMBench (Liu et al., 2024) and GPQA (Rein et al., 2024). While these benchmarks assess broad vision-language understanding, their questions and visual inputs do not reflect agentic decision-making scenarios. They focus on general image understanding and action classification in videos. Recent works (Yao et al., 2024; Yehudai et al., 2025; Wong et al., 2025) show that performance on general-purpose VQA benchmarks does not reliably correlate with success in agentic tasks.

To address this need, several domain-specific benchmarks have advanced the evaluation of agentic capabilities within their respective settings (Koh et al., 2024; Rawles et al., 2024; Bonatti et al., 2025; Jang et al., 2024; Yang et al., 2025b). However, existing benchmarks often provide narrow domain coverage, inputs and action spaces that do not reflect real agentic scenarios, or evaluation sets that are either saturated or affected by noisy annotations.

We introduce **AgentVQA**, a manually curated benchmark for systematically evaluating general-purpose VLMs on realistic agentic scenarios. AgentVQA contains 1,000 single-image, multi-frame, and video-based multiple-choice questions curated from 23 datasets across five domains: **Web Agents**, **Egocentric Videos**, **Robotics**, **Games**, and **Spatial Planning**. These domains represent distinct forms of agent–environment interaction, including GUI navigation, temporally extended first-person understanding, embodied perception and action, strategic decision-making, and 3D spatial planning, as illustrated in Figure 1. By evaluating these settings within a common benchmark, AgentVQA supports both aggregate model comparison and analysis of whether agentic capabilities generalize across domains.

We construct AgentVQA through a human-centered, model-assisted pipeline designed to provide broad coverage, standardized evaluation, and meaningful difficulty. VLMs screen candidates and propose hard distractors, while annotators select, repair, and verify every retained example. Only about 22% of the initial candidates survive the full curation pipeline, as human review identifies subtle ambiguities and annotation errors missed by automated checks and removes examples that are insufficiently challenging. The final benchmark expresses heterogeneous targets, including coordinates, actions, spatial relations, and predicted states, through a common multiple choice interface while preserving the semantics of the original tasks.

Model rankings on AgentVQA correlate with those from corresponding online evaluations, supporting its use as a scalable and reproducible complement to interactive evaluation. At the same time, its rankings differ substantially from those on broad visual and reasoning benchmarks such as MMMU-Pro, GPQA-Diamond, and ChartQAPro, suggesting that performance on conventional benchmarks does not reliably predict agentic competence. Across the 22 open- and closed-source models we evaluate, the strongest model achieves only 58.5% accuracy. Moreover, the strongest model differs across domains, showing that aggregate performance can obscure substantial variation in domain-specific capabilities.

Our cross-domain error analysis reveals a shared limitation that manifests differently across agentic settings. Failures are dominated by grounded scene and state understanding: models struggle with UI localization in **Web Agents**, motion interpretation in **Egocentric Videos**, 3D spatial reasoning in **Robotics** and **Spatial Planning**, and visual state comprehension in **Games**. In contrast, failures arising from misunderstanding the task or selecting an answer inconsistent with the model's

reasoning are comparatively rare. These results suggest that current VLMs often understand the requested objective and can reliably reason over perceived inputs but fail to construct and maintain the grounded representation of the environment needed to act correctly. We release all data and code.

2 Related Work

VLM as agents. Recent advances in multimodal AI have been driven by the development of Vision-Language Models (VLMs). These models, such as QwenVL (Bai et al., 2023), GPT-4o (Hurst et al., 2024), and Gemini (Deepmind, 2026), are typically pretrained on vast, web-scale datasets of paired images and text. This equips them with a broad general-purpose understanding of both visual concepts and natural language, making them an ideal compute unit for agents. Several works have successfully adapted VLMs into specific domains, either by fine-tuning or scaffolding pre-trained models (Black et al., 2024; Koh et al., 2024; Zhou et al., 2025; Zhang et al., 2025; Sarch et al., 2025). AgentVQA evaluates agentic capabilities across diverse domains.

General visual question answering. A wealth of benchmarks have been developed to assess the general capabilities of VLMs, including SEED-Bench (Li et al., 2023a), MMBench (Liu et al., 2024), or MMMU (Yue et al., 2025). While invaluable for measuring core competencies and reasoning, they are ill-suited as proxies for agentic intelligence. Agentic tasks often require episodic memory (Majumdar et al., 2024) because either the modalities are iterative, e.g., a stream of image observations, or the task requires interactive reasoning, e.g., tool-calling. Yao et al. (2024) found that GPT-4o, despite excelling on traditional benchmarks, achieves a failure rate less than 50% on realistic customer service tasks requiring multi-turn interactions. In Fig. 6, we confirm this observation by the divergence between rankings in AgentVQA vs MMMU-Pro and GPQA-Diamond.

Agentic benchmarks. The agentic evaluation landscape spans language-only benchmarks, e.g., SWE-bench (Jimenez et al., 2023) and AgentBench (Liu et al., 2023), which test capabilities such as code generation and tool use, and vision-language benchmarks requiring visual grounding and spatial reasoning. These benchmarks can also be categorized as online or offline. Online benchmarks, e.g., (Savva et al., 2019; Fan et al., 2022; Li et al., 2023b; Koh et al., 2024; Zhang et al., 2025), evaluate agents through environment interaction and therefore capture non-deterministic dynamics important for real-world deployment, but are often costly and infrastructure-intensive (Henderson et al., 2018; Du et al., 2024). Offline benchmarks, e.g., (Rawles et al., 2023; Yang et al., 2025a; Team et al., 2025; Tong et al., 2025; Gong et al., 2025), instead use verifiable question-answer tasks paired with visual inputs. However, existing offline benchmarks are largely domain-specific and differ in answer formats, option difficulty, and sample distributions, making direct aggregation and cross-domain analysis difficult (Table 4). AgentVQA is curated by filtering and transforming samples into a standardized, unambiguous multiple choice format with hard negatives.

3 The AgentVQA Benchmark

We define agentic tasks as those that evaluate action selection or the perceptual, spatial, temporal, and reasoning capabilities required for goal-directed interaction with an environment Bengio et al. (2025). To support scalable offline evaluation of these capabilities, AgentVQA comprises 1,000 manually curated multiple-choice questions drawn from 23 datasets spanning five agentic domains. Its construction is guided by three principles: **comprehensive** coverage across domains to assess skill generality; **standardized** evaluation using manually curated questions with unambiguous ground truth; and **challenging** examples, states, and negative options designed to expose fine-grained weaknesses in current models.

To provide comprehensive coverage, AgentVQA spans five distinct forms of agent-environment interaction: digital interaction and GUI navigation in **Web Agents**; embodied perception and action in **Robotics**; temporally extended first-person planning in **Egocentric Videos**; strategic decision-making in **Games**; and spatial navigation and planning in **Spatial Planning**. As summarized in Figure 2, the benchmark is balanced across domains, with 200 questions per domain. Its inputs range from individual screenshots and embodied-scene images to multi-frame trajectories, egocentric videos, and gameplay videos. Heterogeneous targets, including coordinates, actions, spatial relations, and predicted states, are represented through a common multiple-choice format while preserving observations of the agentic scenario. Full details of the datasets are in Appendix B

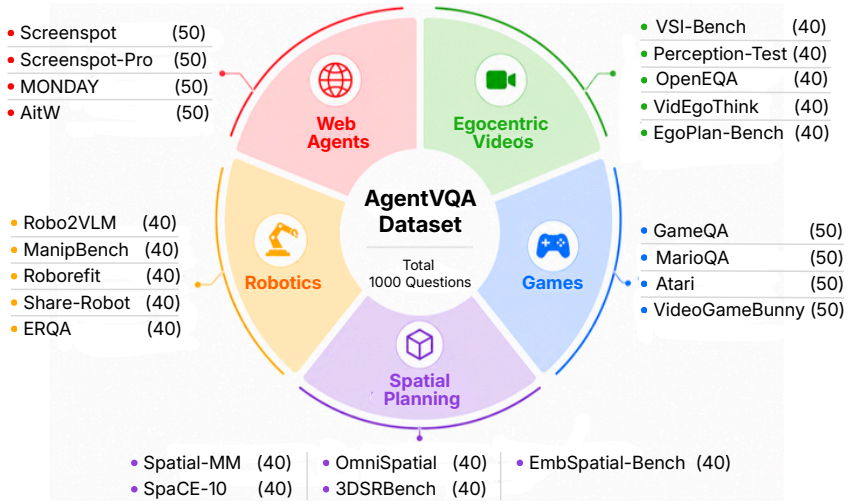


Figure 2: **Distribution of the 1000 questions in the AgentVQA benchmark.** Distribution across the 23 source datasets (sample counts in parentheses), organized within the 5 core agentic domains.

3.1 Data Curation and Construction

We construct AgentVQA through a five-stage, human-centered pipeline. VLMs assist with large-scale screening and hard-negative generation, while human annotators determine which examples are retained and verify the correctness, clarity, and difficulty of every final question.

Stage 1: Candidate collection. We assemble a diverse candidate pool from 23 source datasets spanning five agentic domains. Candidates are sampled across datasets, task types, and input modalities to cover a broad range of realistic observations, states, and decisions, while initially retaining their original questions, annotations, and answer formats.

Stage 2: MCQ construction. We convert the candidate tasks into a common multiple-choice format. For examples that are not originally multiple choice, Gemini 2.5 Pro proposes three hard negative options using the visual input, task prompt, correct answer, action history, and relevant task metadata. The generated distractors include both near-miss negatives, which differ from the correct answer in a fine-grained way, and semantically similar negatives, which correspond to plausible but incorrect objects, actions, or locations.

Stage 3: VLM-assisted screening. We use a multi-turn verification pipeline with Gemini and Claude to identify converted questions with ambiguous instructions, inconsistent annotations, multiple plausible answers, malformed states, implausible distractors, or other quality issues. This high-recall screening removes approximately 45% of the initial candidate pool, reducing the amount of manual review required while preserving human judgment for cases involving subtle visual or domain-specific ambiguities.

Stage 4: Manual curation and repair. Human annotators then inspect the remaining candidates for correctness, difficulty, diversity, and relevance to agentic decision-making. Approximately 60% of the samples that pass automated screening are still rejected because they contain subtle ambiguities, annotation errors, invalid distractors, or are insufficiently challenging. When possible, annotators repair promising examples by refining the question, ground-truth answer, visual input, answer choices, or associated metadata before retaining them for the final evaluation format.

Stage 5: Human verification and benchmark assembly. Every completed MCQ is manually reviewed a second time to ensure that exactly one answer is correct and that the distractors are plausible, challenging, and unambiguously incorrect. This step along with Manual Curation is especially important for grounding tasks, where bounding boxes and automatic spatial checks can admit false positives or reject valid interaction points, as illustrated in Figure 4. We revise questions, annotations, and options as needed, then assemble a balanced final benchmark of 1,000 manually verified questions, with 200 examples per domain and 40 or 50 examples from each source dataset.

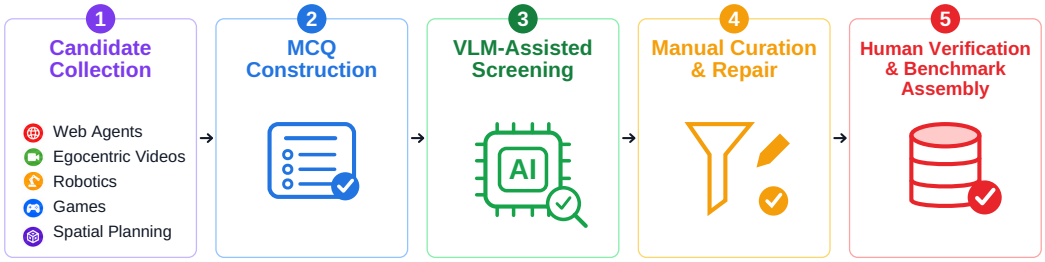


Figure 3: **Dataset curation pipeline for AgentVQA.** (1) Collect candidates across 23 datasets, task types, and input modalities spanning five agentic domains. (2) Standardize tasks as MCQs; Gemini 2.5 Pro proposes three hard negatives when needed. (3) Gemini and Claude flag ambiguity, annotation errors, and weak distractors, removing approximately 45% of candidates. (4) Human annotators review, edit, and repair the remaining samples, rejecting approximately 60% as ambiguous, incorrect, or too easy. (5) Verify exactly one correct answer and plausible, unambiguously incorrect hard negatives, then assemble 1,000 manually verified questions balanced across the five domains.



(a) Bounding-box ambiguity



Based on the screenshot, what is the correct next action to perform to delete the Lyft account?

A CLICK at [280, 958]
B CLICK at [176, 652]
C CLICK at [355, 955]
D Open app drawer

(b) Next-action ambiguity

Figure 4: **Ambiguities addressed by manual curation (Stage 4, Section 3.1).** (a) A rectangular bounding box around scissors includes background, allowing clicks that miss the object to be scored as correct. (b) For the goal of deleting a Lyft account, B (green) directly opens Lyft and is the intended next action. However, D can also lead to a valid solution by opening the app drawer and then launching Lyft, adding a navigation step. The question does not require the shortest route, leaving multiple plausible next actions. Manual review removes or repairs such ambiguous samples to ensure a unique correct answer.

4 Experiments

In this section, we investigate the ability of different models to perform agentic visual tasks across the diverse inputs curated in AgentVQA. We first describe the evaluation setup and model suite in Section 4.1. We then report overall model performance across domains, domain-level trends, action-history effects, open- versus closed-source models, and reasoning variants in Section 4.2, compare AgentVQA rankings with existing offline benchmarks in Section 4.3, and assess its reliability as a lower-cost alternative to online agent evaluation through a comparison with SpatialWorld in Section 4.4. Finally, we examine the failure modes in Section 4.5.

4.1 Experimental Setup

Models. We evaluate a diverse suite of 22 prominent VLMs to ensure a comprehensive assessment of the current landscape. Our selection includes closed-source models: GPT-5.6-sol thinking-high/none (OpenAI, 2026), GPT-5.6-luna thinking-high/none (OpenAI, 2026), GPT-4o (Hurst et al., 2024), Gemini 3.1 Pro (Google, 2026) and 3.7 Flash (Deepmind, 2026), and Claude Opus 4.8 (Anthropic, 2026a) and Sonnet 5 (Anthropic, 2026b). We also include a wide range of open-source models: the GLM series (4.6V (Z.AI, 2025), 4.5V (GLM-V Team, 2026), 4.1V-9B-Thinking/Base (GLM-V Team, 2026)), Kimi K3 (Team, 2026), Llama 4 Maverick (Meta, 2025),

the Qwen3.5 series (397B-A17B, 122B-A10B, 35B-A3B, 27B) (Qwen Team, 2026a), Qwen3.8 27B (Qwen Team, 2026c) and the Qwen3.7 Plus and Flash variants (Qwen Team, 2026b).

Models are categorized as "Reasoning" or "Non-Reasoning" based on whether they allocate extended token budgets for internal chain-of-thought reasoning before producing a final answer.

Evaluation setup. For models that do not natively support video input, we sample 32 frames at evenly spaced intervals. This number is validated by our own ablation studies and prior work (Yang et al., 2025a). Models are evaluated with a temperature of 0.8 and 0.2 for reasoning and non-reasoning models, respectively. For models without manual temperature, we use the default.

We use a standard prompt that provides the question and any relevant context, querying the model to output only the single letter corresponding to its choice. For questions involving spatial coordinates, options are formatted in each model's native convention (e.g., normalized [0, 1000] in (y, x) order for Gemini, absolute pixel coordinates for Qwen). Final answers are extracted programmatically, and all visual inputs are provided at native resolution. Further details on prompts and answer extraction are in Appendices K and E.

Table 1: **Performance of VLMs on AgentVQA across the five agentic domains.** 🤔 represents Reasoning Models. 🔒 represents closed-source models. Bolded and underlined numbers indicate best and second-best rank per column, respectively.

Model	Web Agents	Egocentric Videos	Games	Spatial Planning	Robotics	Avg
GPT 5.6-sol think-high 🤔🔒	**47.0**	<u>69.0</u>	62.5	51.0	45.5	55.0
GPT 5.6-sol think-none 🔒	16.0	59.0	44.5	41.0	30.5	38.2
GPT 5.6-luna think-high 🤔🔒	34.5	65.0	61.0	43.5	35.5	47.9
GPT 5.6-luna think-none 🔒	19.5	46.0	37.5	32.0	29.0	32.8
GPT-4o 🔒	21.5	40.0	29.0	35.0	31.5	31.4
Gemini 3.1-pro 🤔🔒	34.5	64.0	<u>68.0</u>	<u>61.0</u>	<u>51.0</u>	<u>55.7</u>
Gemini 3.7-flash 🔒	40.5	67.0	**70.0**	**62.5**	**52.5**	**58.5**
Claude Opus 4.8 🔒	35.5	60.0	50.5	41.0	33.0	44.0
Claude Sonnet 5 🔒	38.0	47.0	53.0	42.5	36.5	43.4
GLM 4.6v 🤔	25.5	54.0	47.5	39.5	42.5	41.8
GLM 4.5v 🤔	25.0	65.0	46.5	41.0	37.0	42.9
GLM 4.1V 9B-Thinking 🤔	19.0	53.0	41.0	38.5	25.5	35.4
GLM 4.1V 9B-Base	20.5	46.0	37.5	44.5	33.0	36.3
Kimi K3 🤔	22.0	64.0	63.5	53.0	39.0	48.3
Llama-4-Maverick	18.5	49.0	23.0	30.0	19.5	28.0
Qwen3.8 27B 🤔	30.0	<u>69.0</u>	61.5	51.5	42.5	50.9
Qwen3.7 Plus 🤔	32.5	60.0	61.5	50.0	39.0	48.6
Qwen3.7 Flash 🤔	<u>43.0</u>	**72.5**	61.0	55.0	43.5	55.0
Qwen3.5 397B-A17B 🤔	25.0	65.0	57.5	51.5	40.0	47.8
Qwen3.5 122B-a10b 🤔	26.0	60.0	53.0	50.0	41.0	46.0
Qwen3.5 35B-a3b 🤔	34.5	62.0	55.5	45.5	42.0	47.9
Qwen3.5 27B 🤔	32.5	57.0	49.5	51.5	43.0	46.7

4.2 Results on AgentVQA

We report the results on AgentVQA in Table 1. We draw the following conclusions:

Top models struggle with AgentVQA. Even the top performing model, Gemini 3.7 Flash, achieves only 58.5% accuracy. A clear tiering is observable: the largest proprietary models (GPT-5.6, Gemini 3.7 Flash, Gemini 3.1 Pro) are at the top, followed by large open-source models such as Qwen3.7 Flash, with smaller and older-generation models in the lower tiers. Among the five domains, models perform best on Egocentric Videos and Games, followed by Spatial Planning, Robotics, and worst on Web Agents. This indicates that current VLMs are stronger at temporally contextualized visual understanding and game-state reasoning than at precise GUI grounding and spatial reasoning.

Large variance per domain. Current VLMs are strongest on temporally grounded understanding (Egocentric Videos: 59%, Games: 51.5%) and weaker on precise grounding and embodied reasoning (Spatial Planning: 46%, Robotics: 38%, Web Agents: 29%). Web Agents is the hardest domain, with only three models exceeding 40%. Games exhibits the largest spread (23% to 70%), suggesting strategic reasoning is strong only in the most capable models. No single model dominates across all domains: Gemini 3.7 Flash leads Games (70%) but falls behind GPT-5.6-sol thinking-high in Web Agents (40.5% vs. 47%), while Qwen3.7 Flash leads Egocentric Videos (72.5%) and Gemini 3.7 Flash leads Spatial Planning (62.5%) and Robotics (52.5%). These cross-domain rank shifts indicate highly uneven capabilities across perceptual, spatial, and interactive settings.

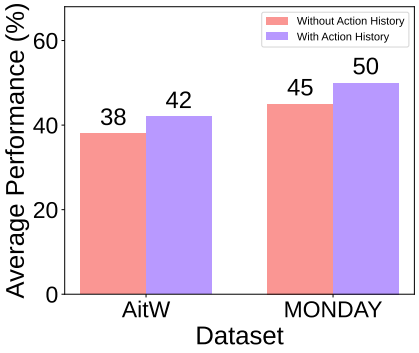


Figure 5: Impact of action history on model performance in Web Agents.

Open-source models are catching up. The gap between closed-source and open-source models remains narrow. The top closed-source model, Gemini 3.7 Flash (58.5%), leads the top open-source model, Qwen3.7 Flash (55.0%), by 3.5%. Kimi K3 (48.3%), Qwen3.5 35B (47.9%), and Qwen3.5 397B (47.8%) outperform Claude Opus 4.8 (44.0%) and Sonnet 5 (43.4%). Multiple open-source models also substantially outperform GPT-5.6-sol thinking-none (38.2%) and GPT-4o (31.4%). These results suggest that while the strongest proprietary models retain an advantage on agentic tasks, competitive open-source alternatives already match many closed-source systems.

Reasoning offers varying results. Extended reasoning helps some models but not others. GPT-5.6-sol gains 16.8 points with thinking-high, while GLM 4.1V performs worse with thinking as compared to without it. We find that GLM-4.1V-Thinking gets stuck in "thinking loops" (App. J).

Trajectory questions benefit from action history. Our ablation on Web Agents tasks reveals that providing action history is a consistent benefit (Figure 5). Across five models, removing the trajectory history decreases average performance by 9.5% on AitW and 10% on MONDAY. This confirms that for multi-step tasks, access to historical context is critical for decision-making.

4.3 Rank Correlation

Low correlation with existing benchmarks. To validate the benefit of a specialized agentic benchmark, we compare model rankings for a subset of models on AgentVQA to those on a subset of MMMU-Pro (Yue et al., 2025), ChartQA-Pro (Masry et al., 2025), and GPQA-Diamond (Rein et al., 2024), running our own evaluations with standardized prompts. As shown in Figure 6, rankings diverge substantially. Spearman's rank correlation coefficients are moderate for MMMU-Pro ($\rho = 0.67$) and GPQA-Diamond ($\rho = 0.59$), and weak for ChartQA-Pro ($\rho = 0.44$).

No single model dominates across all comparisons. Gemini 3.1 Pro ranks second on AgentVQA and MMMU-Pro but drops to 5th on GPQA-Diamond, while Qwen3.8 27B leads GPQA-Diamond but falls to 6th on MMMU-Pro. These shifts confirm that strong performance on general-purpose visual reasoning does not reliably predict agentic competence.

4.4 Correlation with Online Agent Evaluations

To examine whether AgentVQA provides a reliable alternative to costly interactive evaluation, we compare its model rankings with performance on SpatialWorld (Gao et al., 2026), a closed-loop benchmark for embodied and interactive agents. SpatialWorld scores are obtained from the official released evaluation results, which are produced under a single simulator-agnostic protocol across eight simulation backends. We evaluate the same models on AgentVQA, and compare their relative rankings via Spearman's rank correlation. This analysis measures whether AgentVQA preserves the model ordering observed under interactive evaluation despite operating entirely offline and avoiding environment configuration and execution-related computational costs.

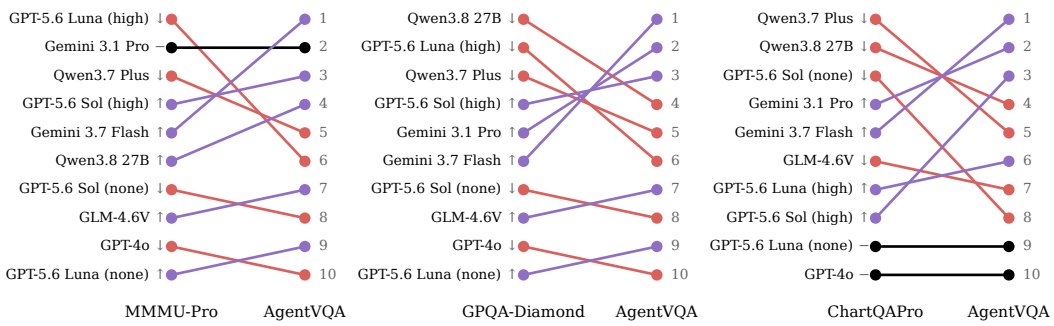


Figure 6: **Divergence in model performance rankings between AgentVQA and three existing benchmarks.** Lines are colored red if a model's rank degrades and purple if it improves. The rank divergence shows that our benchmark is largely uncorrelated with general VQA and domain-specific agentic benchmarks. While a few models maintain their ranks, the relative positions of most other models change. In the comparison with GPQA-Diamond, no model maintains its relative ranking.

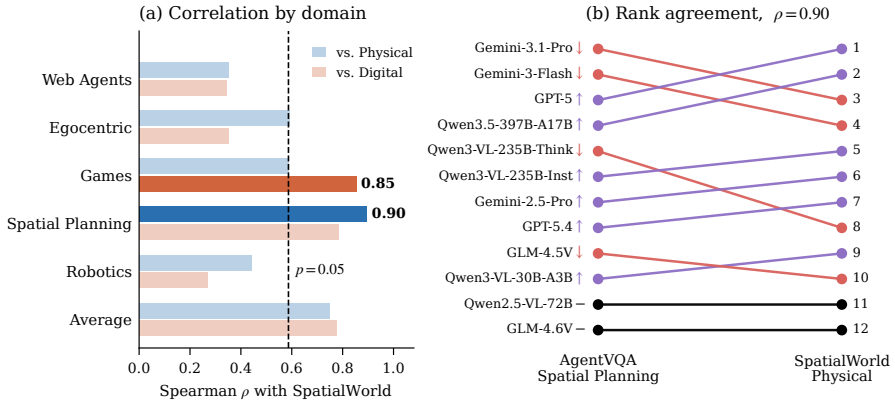


Figure 7: **AgentVQA ranks models the way an online interactive benchmark does.** (a) Spearman correlation between each AgentVQA domain and the two SpatialWorld tracks over the $n = 12$ shared models; saturated bars are the domain-matched pairs. (b) Model rankings under AgentVQA Spatial Planning against the SpatialWorld Physical track ($\rho = 0.895$, $p = 8 \times 10^{-5}$). **Purple** marks models that rank higher on SpatialWorld, **red** for models lower on it, and black unchanged.

Figure 7 shows a strong positive Spearman rank correlation of $\rho = \mathbf{0.895}$ across **12** overlapping models between SpatialWorld and AgentVQA Spatial Planning, the closest category. This agreement demonstrates that AgentVQA provides a reliable and substantially lower-cost offline alternative for ranking agentic models, avoiding the environment setup, repeated rollouts, and computational expense required by online evaluation. The lower correlations for other AgentVQA domains suggest that model rankings depend on the capabilities being assessed, highlighting the limitations of a single dataset as a proxy for overall agentic performance. Together with the strong agreement in the matched domain, this pattern supports the interpretation that AgentVQA captures meaningful domain-specific performance differences.

4.5 Error Modes

To understand why models fail, we analyze incorrect predictions from Gemini 3.1 Pro, sampled approximately evenly across datasets. We prompt the model to provide its reasoning with each answer, manually establish a taxonomy of common failure modes, use Gemini to categorize the errors, and manually verify the resulting assignments.

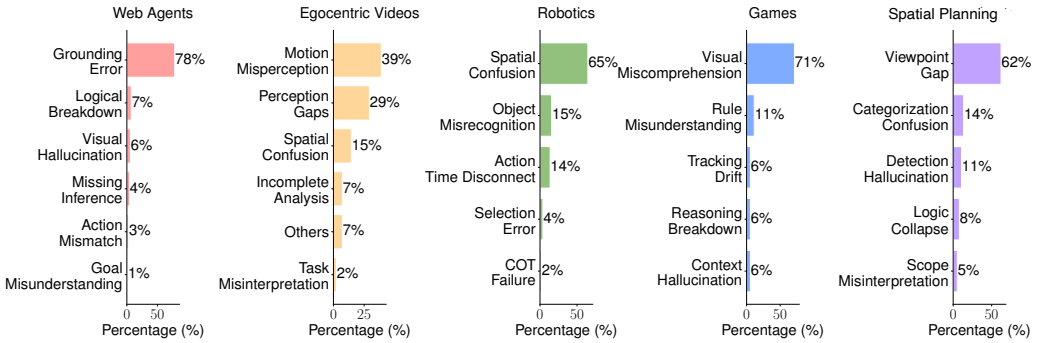


Figure 8: **Distribution of the most common error modes across the five agentic domains in AgentVQA.** The analysis is based on a semi-automated categorization of the incorrect predictions from Gemini-3.1-Pro. The chart is organized by domain, with each colored segment representing a primary domain. Each slice within a segment corresponds to a specific error mode, and its size reflects its prevalence (%).

The results reveal domain-specific manifestations of a shared weakness: constructing an accurate task-relevant state from visual observations (Figure 8). In **Web Agents**, grounding errors dominate (78%), indicating difficulty locating the correct interface element. In **Games**, visual miscomprehension (71%) shows that failures primarily arise from reading the game state rather than applying its rules. **Robotics** and **Spatial Planning** are dominated by spatial confusion (65%) and viewpoint gaps (62%), respectively, exposing weaknesses in relational and perspective-aware reasoning. Errors in **Egocentric Videos** are more distributed across motion misperception (39%), perception gaps (29%), and spatial confusion (15%), reflecting difficulty tracking dynamic states. Across domains, failures therefore arise mainly before high-level planning, during perception, grounding, and state construction. Additional details are provided in Appendix G.

5 Conclusion

We introduce AgentVQA, a unified offline benchmark that consolidates 23 datasets across five agentic domains into 1,000 manually verified multiple-choice questions with challenging hard negatives. While prior work typically evaluates agents within individual environments or domains (Koh et al., 2024; Rawles et al., 2024; Bonatti et al., 2025; Yang et al., 2025b), AgentVQA enables controlled comparison of visual decision-making capabilities across diverse agentic settings.

Our results show that these capabilities remain far from solved: among 22 state-of-the-art VLMs, the strongest model achieves only 58.5% accuracy. Consistent with findings from domain-specific agent evaluations, errors arise primarily from grounding, perception, motion understanding, and spatial state construction rather than high-level reasoning alone. Action history improves performance, while increased inference-time reasoning benefits some models but not others. However, neither closes the remaining gap. Moreover, AgentVQA rankings diverge from traditional VQA benchmarks while correlating strongly with SpatialWorld, suggesting that conventional benchmarks do not fully capture agentic readiness and that AgentVQA offers a reliable, substantially lower-cost alternative for rapid agent-model evaluation.

Taken together, these findings support viewing agentic competence as a collection of domain-specific capabilities rather than a single general ability. The strong agreement with online evaluation in the matched domain, alongside weaker cross-domain agreement, suggests that AgentVQA can help distinguish these capabilities and guide more targeted model development. Within the broader evaluation landscape, controlled offline benchmarks can complement interactive environments by identifying weaknesses efficiently and directing subsequent online testing toward the domains and failure modes that require closer investigation.

Future work should connect specific offline error patterns to failures during interactive execution. Such analyses could guide improvements in grounding and state tracking while testing whether gains on controlled questions translate into more reliable online behavior.

6 AI USE STATEMENT

In accordance with the ICLR 2027 policy on large language model usage, we disclose all substantive uses of LLMs in our research pipeline.

Hard negative generation. We use Gemini 2.5 Pro to generate hard negative distractor options for MCQ conversion (Section 3.1). The selection was based on a preliminary comparison against GPT-4o and Qwen 2.5 VL (7B). The exact prompt is in Appendix L.

Automated sample filtering. Prior to manual curation, we use a multi-turn pipeline with Gemini and Claude to flag potentially ambiguous questions, inconsistent annotations, multiple plausible answers, malformed states, implausible distractors, and other quality issues. These automated checks serve only as a screening stage; retained examples are subsequently reviewed, repaired when necessary, and verified by human annotators as described in Section 3.1.

Error-mode annotation. For the error analysis in Section 4.5 and Appendix G, we use Gemini 3.1 Pro to assign predefined error-mode categories to incorrect model predictions. The taxonomy is established through manual analysis, and the model is provided with the visual input, question, answer choices, ground-truth answer, incorrect prediction, and model reasoning when available. A randomly selected subset of the resulting automated assignments is manually verified by the authors. The annotation prompt is provided in Appendix M.

Writing Assistance. We use LLMs to improve readability and clarity of text.

All AI-assisted content has been reviewed by the authors and we take full responsibility for the final content of this work.

7 ETHICS STATEMENT

AgentVQA draws on existing datasets and human annotation. Its use should respect source-data licenses, attribution requirements, privacy, and annotator welfare. Benchmark performance should not be interpreted as evidence that a model is safe or reliable for autonomous deployment.

8 REPRODUCIBILITY STATEMENT

To support reproducibility, Sections 3–4 and the appendix describe dataset construction, annotation, and evaluation procedures. All data, evaluation protocols, and code will be released upon acceptance.

REFERENCES

Anthropic. Claude Opus 4.8 System Card. https://www-cdn.anthropic.com/0b4915911bb0d19eca5b5ee635c80fef830a37ea.pdf, May 2026a.

Anthropic. Claude Sonnet 5 System Card. https://www-cdn.anthropic.com/283ef97c476cf442c91d9a37d5b214242a55bb92/Claude%20Sonnet%205%20System%20Card.pdf, June 2026b.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions. *arXiv preprint arXiv:2505.23281*, 2025.

Yoshua Bengio, Michael Cohen, Damiano Fornasiere, Joumana Ghosn, Pietro Greiner, Matt MacDermott, Sören Mindermann, Adam Oberman, Jesse Richardson, Oliver Richardson, Marc-Antoine Rondeau, Pierre-Luc St-Charles, and David Williams-King. Superintelligent agents pose catastrophic risks: Can scientist ai offer a safer path?, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, brian ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 27–30 Sep 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026.

Rogerio Bonatti, Dan Zhao, Francesco Bonacci, Dillon Dupont, Sara Abdali, Yinheng Li, Yadong Lu, Justin Wagle, Kazuhito Koishida, Arthur Bucker, Lawrence Keunho Jang, and Zheng Hui. Windows agent arena: Evaluating multi-modal OS agents at scale. In *Forty-second International Conference on Machine Learning*, 2025.

Kaiyuan Chen, Shuangyu Xie, Zehan Ma, Pannag R Sanketi, and Ken Goldberg. Robo2vlm: Visual question answering from large-scale in-the-wild robot manipulation datasets. *arXiv preprint arXiv:2505.15517*, 2025.

Yi Chen, Yuying Ge, Yixiao Ge, Mingyu Ding, Bohao Li, Rui Wang, Ruifeng Xu, Ying Shan, and Xihui Liu. Egoplan-bench: Benchmarking multimodal large language models for human-level planning. *arXiv preprint arXiv:2312.06722*, 2023.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *ACL*, 2024a.

Sijie Cheng, Kechen Fang, Yangyang Yu, Sicheng Zhou, Bohao Li, Ye Tian, Tingguang Li, Lei Han, and Yang Liu. Videgothink: Assessing egocentric video understanding capabilities for embodied ai. *arXiv preprint arXiv:2410.11623*, 2024b.

Sijie Cheng, Zhicheng Guo, Jingwen Wu, Kechen Fang, Peng Li, Huaping Liu, and Yang Liu. Egothink: Evaluating first-person perspective thinking capability of vision-language models. In *CVPR*, 2024c.

Sudeep Dasari, Jianren Wang, Joyce Hong, Shikhar Bahl, Yixin Lin, Austin Wang, Abitha Thankaraj, Karanbir Chahal, Berk Calli, Saurabh Gupta, et al. Rb2: Robotic manipulation benchmarking with a twist. *arXiv preprint arXiv:2203.08098*, 2022.

Google Deepmind. Gemini 3.7 Flash System Card. https://deepmind.google/models/model-cards/gemini-3-7-flash/, August 2026.

Mengfei Du, Binhao Wu, Zejun Li, Xuanjing Huang, and Zhongyu Wei. Embspatial-bench: Benchmarking spatial understanding for embodied tasks with large vision-language models. *arXiv preprint arXiv:2406.05756*, 2024.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35, 2022.

Hongcheng Gao, Hailong Qu, Jingyi Tang, Jiahao Wang, Zihao Huang, Hengkang Qiao, Shihong Huang, Junming Yang, Yi Li, Hongyixuan Yuan, Wenjie Li, Bohan Zeng, Wenbo Li, Bo Wang, Jianhui Liu, Olive Huang, Haoyang Huang, Wentao Zhang, Guoqing Huang, Nan Duan, and

Yinpeng Dong. Spatialworld: Benchmarking interactive spatial reasoning of multimodal agents in real-world tasks, 2026.

GLM-V Team. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning, 2026.

Ziyang Gong, Wenhao Li, Oliver Ma, Songyuan Li, Jiayi Ji, Xue Yang, Gen Luo, Junchi Yan, and Rongrong Ji. Space-10: A comprehensive benchmark for multimodal large language models in compositional spatial intelligence. *arXiv preprint arXiv:2506.07966*, 2025.

Google. Gemini 3.1 Pro Model Card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-1-Pro-Model-Card.pdf, February 2026.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *CVPR*, 2024.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. *arXiv preprint arXiv:2109.06780*, 2021.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *AAAI*, volume 32, 2018.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Lawrence Jang, Yinheng Li, Dan Zhao, Charles Ding, Justin Lin, Paul Pu Liang, Rogerio Bonatti, and Kazuhito Koishida. Videowebarena: Evaluating long context multimodal agents with video understanding web tasks. *arXiv preprint arXiv:2410.19100*, 2024.

Yunseok Jang, Yeda Song, Sungryull Sohn, Lajanugen Logeswaran, Tiange Luo, Dong-Ki Kim, Kyunghoon Bae, and Honglak Lee. Scalable video-to-dataset generation for cross-platform mobile agents. In *CVPR*, 2025.

Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, Xinda Xue, Qinghang Su, Huaihai Lyu, Xiaolong Zheng, Jiaming Liu, Zhongyuan Wang, and Shanghang Zhang. Robobrain: A unified brain model for robotic manipulation from abstract to concrete, 2025.

Mengdi Jia, Zekun Qi, Shaochen Zhang, Wenyao Zhang, Xinqiang Yu, Jiawei He, He Wang, and Li Yi. Omnispatial: Towards comprehensive spatial reasoning benchmark for vision language models, 2026.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. *arXiv preprint arXiv:2210.03094*, 2(3), 2022.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*, 2023.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*, 2024.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023a.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *CoRL*. PMLR, 2023b.

Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. Screenspot-pro: Gui grounding for professional high-resolution computer use. *arXiv preprint arXiv:2504.07981*, 2025.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*, 2023.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *ECCV*. Springer, 2024.

Yuhao Lu, Yixuan Fan, Beixing Deng, Fangfu Liu, Yali Li, and Shengjin Wang. Vl-grasp: a 6-dof interactive grasp policy for language-oriented objects in cluttered indoor scenes. In *IROS*. IEEE, 2023.

Wufei Ma, Haoyu Chen, Guofeng Zhang, Yu-Cheng Chou, Celso M de Melo, and Alan Yuille. 3dsrbench: A comprehensive 3d spatial reasoning benchmark. *arXiv preprint arXiv:2412.07825*, 2024.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *CVPR*, 2024.

Ahmed Masry, Mohammed Saidul Islam, Mahir Ahmed, Aayush Bajaj, Firoz Kabir, Aaryaman Kartha, Md Tahmid Rahman Laskar, Mizanur Rahman, Shadikur Rahman, Mehrad Shahmohammadi, Megh Thakkar, Md Rizwan Parvez, Enamul Hoque, and Shafiq Joty. Chartqapro: A more diverse and challenging benchmark for chart question answering, 2025.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3), 2022.

Meta. The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation. https://ai.meta.com/blog/llama-4-multimodal-intelligence/, 2025.

Stephanie Milani, Anssi Kanervisto, Karolis Ramanauskas, Sander Schulhoff, Brandon Houghton, and Rohin Shah. Bedd: The minerl basalt evaluation and demonstrations dataset for training and benchmarking agents that solve fuzzy tasks. *Advances in Neural Information Processing Systems*, 36:32867–32878, 2023.

Jonghwan Mun, Paul Hongsuck Seo, Ilchae Jung, and Bohyung Han. Marioqa: Answering questions by watching gameplay videos. In *ICCV*, 2017.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

OpenAI. GPT-5.6 System Card. https://deploymentsafety.openai.com/gpt-5-6, July 2026.

Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wolczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, et al. Balrog: Benchmarking agentic llm and vlm reasoning on games. *arXiv preprint arXiv:2411.13543*, 2024.

Viorica Patraucean, Lucas Smaira, Ankush Gupta, Adria Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Mateusz Malinowski, Yi Yang, Carl Doersch, et al. Perception test: A diagnostic benchmark for multimodal video models. *Advances in Neural Information Processing Systems*, 36, 2023.

Aske Plaat, Max van Duijn, Niki van Stein, Mike Preuss, Peter van der Putten, and Kees Joost Batenburg. Agentic large language models, a survey, 2025.

Qwen Team. Qwen3.5. https://qwen.ai/blog?id=qwen3.5, February 2026a.

Qwen Team. Qwen3.7. https://qwen.ai/blog?id=qwen3.7, May 2026b.

Qwen Team. Qwen3.8. https://qwen.ai/blog?id=qwen3.8, August 2026c.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. Androidinthewild: A large-scale dataset for android device control. *Advances in Neural Information Processing Systems*, 36, 2023.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, et al. Androidworld: A dynamic benchmarking environment for autonomous agents. *arXiv preprint arXiv:2405.14573*, 2024.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent, 2022.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *COLM*, 2024.

Gabriel Sarch, Balasaravanan Thoravi Kumaravel, Sahithya Ravi, Vibhav Vineet, and Andrew D Wilson. Grounding task assistance with multimodal cues from a single demonstration. *arXiv preprint arXiv:2505.01578*, 2025.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *ICCV*, 2019.

Fatemeh Shiri, Xiao-Yu Guo, Mona Golestan Far, Xin Yu, Gholamreza Haffari, and Yuan-Fang Li. An empirical analysis on spatial reasoning capabilities of large multimodal models, 2024.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Mohammad Reza Taesiri and Cor-Paul Bezemer. Videogamebunny: Towards vision assistants for video games, 2024.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

Kimi Team. Kimi k3: Open frontier intelligence, 2026.

Jingqi Tong, Jixin Tang, Hangcheng Li, Yurong Mou, Ming Zhang, Jun Zhao, Yanbo Wen, Fan Song, Jiahao Zhan, Yuyang Lu, Chaoran Tao, Zhiyuan Guo, Jizhou Yu, Tianhao Cheng, Changhao Jiang, Zhen Wang, Tao Liang, Zhihui Fei, Mingyang Wan, Guojun Ma, Weifeng Ge, Guanhua Chen, Tao Gui, Xipeng Qiu, Qi Zhang, and Xuanjing Huang. Game-rl: Synthesizing verifiable game tasks at scale to boost vlms general reasoning. *arXiv preprint arXiv:2505.13886*, 2025.

Quan Vuong, Sergey Levine, Homer Rich Walke, Karl Pertsch, Anikait Singh, Ria Doshi, Charles Xu, Jianlan Luo, Liam Tan, Dhruv Shah, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *CoRL*, 2023.

Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, et al. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world. In *ICCV*, 2023.

Ryan Wong, Jiawei Wang, Junjie Zhao, Li Chen, Yan Gao, Long Zhang, Xuan Zhou, Zuo Wang, Kai Xiang, Ge Zhang, et al. Widesearch: Benchmarking agentic broad info-seeking. *arXiv preprint arXiv:2508.07999*, 2025.

Frank F Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, et al. Theagentcompany: benchmarking llm agents on consequential real world tasks. *arXiv preprint arXiv:2412.14161*, 2024.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *CVPR*, 2025a.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, Heng Ji, Huan Zhang, and Tong Zhang. EmbodiedBench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 70576–70631. PMLR, 13–19 Jul 2025b.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, et al. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*, 2025c.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. Tau-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

Asaf Yehudai, Lilach Eden, Alan Li, Guy Uziel, Yilun Zhao, Roy Bar-Haim, Arman Cohan, and Michal Shmueli-Scheuer. Survey on evaluation of llm-based agents. *arXiv preprint arXiv:2503.16416*, 2025.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark, 2025.

Z.AI. Glm-4.6v. https://z.ai/blog/glm-4.6v, December 2025.

Alex L Zhang, Thomas L Griffiths, Karthik R Narasimhan, and Ofir Press. Videogamebench: Can vision-language models complete popular video games? *arXiv preprint arXiv:2505.18134*, 2025.

Ruohan Zhang, Calen Walshe, Zhuode Liu, Lin Guan, Karl Muller, Jake Whritner, Luxin Zhang, Mary Hayhoe, and Dana Ballard. Atari-head: Atari human eye-tracking and demonstration dataset. In *AAAI*, volume 34, 2020.

Enyu Zhao, Vedant Raval, Hejia Zhang, Jiageng Mao, Zeyu Shangguan, Stefanos Nikolaidis, Yue Wang, and Daniel Seita. Manipbench: Benchmarking vision-language models for low-level robot manipulation, 2025.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

Zhongyi Zhou, Yichen Zhu, Junjie Wen, Chaomin Shen, and Yi Xu. Vision-language-action model with open-world embodied reasoning from pretrained knowledge. *arXiv preprint arXiv:2505.21906*, 2025.

Hao Zhu, Raghav Kapoor, So Yeon Min, Winson Han, Jiatai Li, Kaiwen Geng, Graham Neubig, Yonatan Bisk, Aniruddha Kembhavi, and Luca Weihs. Excalibur: Encouraging and evaluating embodied exploration. In *CVPR*, 2023.

A Dataset Selection and Curation Rationale

The construction of AgentVQA is guided by a principled selection process to ensure the final benchmark is comprehensive, robust, and effective for evaluating the agentic capabilities of Vision-Language Models (VLMs). Our domains: **Web Agents**, **Egocentric Videos**, **Robotics**, **Games**, and **Spatial Planning** are chosen to represent a broad spectrum of agentic challenges, from digital interaction to embodied physical perception and strategic planning. The following five criteria are central to our selection of the 23 source datasets.

High agentic signal. A primary requirement is that each dataset's tasks must involve a strong "agentic signal." This means moving beyond passive visual question answering (e.g., "What color is the car?") to scenarios that demand active reasoning from an agent's perspective. We prioritize datasets where the core task involves:

- Choosing actions or predicting the next optimal step.
- Evaluating the consequences of a sequence of actions.
- Understanding causality and the relationships between actions and outcomes.
- Strategic planning toward a specific goal.

This focus ensures that AgentVQA directly probes the decision-making and reasoning capabilities that are fundamental to agentic intelligence.

Comprehensive and distinct domain coverage. To test for generalist agentic behavior, we select datasets that span diverse domains and modalities. Each of the five chosen domains introduces a unique set of challenges and required skills not fully covered by the others:

- **Web Agents**: Tests GUI navigation, grounding, and interaction in a digital environment.
- **Robotics** & **Egocentric Videos**: Evaluate embodied perception, spatial awareness, and understanding of the physical world from a first-person perspective.
- **Games**: Focus on strategic decision-making, reward modeling, and long-term planning in rule-based environments.
- **Spatial Planning**: Assesses foundational spatial awareness and reasoning, a critical component for any embodied agent.

This multi-domain approach prevents the benchmark from being too narrow and allows for a holistic assessment of a VLM's generalization ability.

Offline convertibility. A crucial practical consideration is the feasibility of converting each dataset into a standardized, offline format. This criterion requires that the source data (e.g., trajectories, videos) be sufficiently structured to allow for an unambiguous conversion into a Multiple-Choice Question (MCQ) format. Every resulting MCQ must have a single, verifiably correct answer to enable scalable, deterministic, and efficient evaluation without relying on resource-intensive online simulators.

Public availability and permissive licensing. To ensure transparency, reproducibility, and accessibility for the broader research community, all selected datasets must be publicly available. Furthermore, their licenses must permit modification and redistribution as part of a new benchmark, allowing us to legally and ethically create and share AgentVQA.

Sufficient scale. The selected datasets need to be large enough to provide a diverse pool from which high-quality samples can be curated. A larger source pool increases the likelihood of finding samples that are both challenging and unambiguous across a range of question types. This criterion ensures that our manual curation process (Section 3.1) has adequate coverage of each domain and dataset.

B Source Dataset Overview

Table 2 summarizes the source datasets used in AgentVQA. The input and output columns describe each dataset after conversion into AgentVQA. Every output is represented as a multiple choice question, while the listed output type identifies the meaning of the selected option.

C Hard Negative Generation Pipeline

The difficulty of AgentVQA is ensured by a systematic process for generating challenging distractor options, or "hard negatives." This process transforms tasks that are not inherently multiple-choice, such as web navigation trajectories, into a standardized and rigorous MCQ format. Our pipeline creates negatives that are both plausible and unambiguously incorrect, forcing models to demonstrate fine-grained understanding rather than relying on simple heuristics. The process is detailed below.

VLM-assisted generation. To select the generation model, we first conducted a preliminary study on a data subset using Gemini 2.5 Pro, GPT-4o, and Qwen 2.5 VL (7B). After manually comparing the generated samples for both difficulty and accuracy, we selected Gemini 2.5 Pro for its superior performance. We therefore employ a VLM-assisted pipeline centered around **Gemini 2.5 Pro** to generate hard negatives. For each question, we provide the model with comprehensive context, including the primary image or screenshot, the task prompt, the ground-truth answer, any relevant action history for sequential tasks, and metadata such as image dimensions and coordinate formats (e.g., (x,y) for points, or (x_1, y_1, x_2, y_2) for bounding boxes). For grounding tasks, we also visually render the correct action's location onto the image. Given this context, we prompt the model to generate three distinct hard negatives as a combination of two types: **near-miss negatives** and **semantically similar negatives**. Near-miss negatives are distractors spatially close to the correct answer that test precision, such as coordinates a few pixels away from a correct UI element. Semantically similar negatives are conceptually related to the correct answer and test the model's ability to differentiate between similar objects, such as choosing a "Mobile" phone when the correct answer is a "Tablet." The exact prompt used for generating these "hard negatives" is in Appendix L

Verification and filtering. To ensure quality, the generated negatives undergo a two-stage verification process. First, an automated script filters grounding-based distractors by verifying their coordinates fall outside the ground-truth bounding box, programmatically confirming the option is a "negative." Following this, a subset of these samples undergoes a rigorous manual review. We manually verify each distractor for both plausibility, ensuring the option is a believable choice that could confuse a model, and unambiguous incorrectness, which guarantees there is only one correct answer to the MCQ. This human-in-the-loop step is critical for ensuring the quality of the generated negatives prior to the manual curation stage described in Section 3.1. The hard negatives produced by this pipeline form the initial candidate pool. The final 1000-sample benchmark undergoes additional manual curation (Section 3.1) to verify correctness, difficulty, and the absence of any ambiguity.

D Definition of Agentic Tasks

The community defines agentic tasks as those requiring a system to "observe their environment and act in it in order to achieve goals" Bengio et al. (2025), encompassing perception to sense the state, intelligence to reason and plan, and affordances to execute actions Plaat et al. (2025). While traditional image VQA treats visual understanding as an end in itself, agentic VQA frames perception as a means to action. Here, each answer must inform executable decisions rather than merely describe what is seen Reed et al. (2022). To ensure AgentVQA captures this full capability range, we structure our sub-tasks into a unified hierarchy: Action Grounding tasks (e.g., Tap) test the critical translation of semantic intent into executable spatial actions Reed et al. (2022); Spatiotemporal Reasoning tasks evaluate the dynamic mental maps required for movement; State Understanding tasks (e.g., Functional Reasoning) ensure the agent can accurately parse the environment to inform decision-making; and Strategic Planning tasks (e.g., Reward Modeling) assessing long-term outcomes through multi-step reasoning Sutton et al. (1998).

Table 2: Source datasets used in AgentVQA. Descriptions and modalities follow the released AgentVQA data fields and the original dataset definitions.

Dataset	Description	Input type	Output type
Web Agents			
AitW Rawles et al. (2023)	Android device control from a task goal, the current screen, and previous actions.	Image	Next UI action
MONDAY Jang et al. (2025)	Android and iOS navigation trajectories extracted from instructional videos.	Image	Next UI action
Screenspot Cheng et al. (2024a)	Grounding an instruction to an actionable element in mobile, desktop, or web interfaces.	Image	Click coordinates
Screenspot-pro Li et al. (2025)	Grounding an instruction in high resolution interfaces from professional applications.	Image	Click coordinates
Robotics			
ERQA Team et al. (2025)	Embodied reasoning about robot scenes, actions, trajectories, and task progress.	Multiple images	Reasoning answer
Robo2VLM Chen et al. (2025)	Visual questions derived from real robot manipulation trajectories to test spatial, goal conditioned, and interaction reasoning.	Image	Task state or relation
Roborefit Lu et al. (2023)	Grounding a language specified grasp target in a cluttered indoor scene.	Image	Grasp coordinates
ManipBench Zhao et al. (2025)	Low level manipulation over rigid, articulated, deformable, tool use, and dynamic tasks.	Image	Manipulation trajectory
Share-Robot Ji et al. (2025)	Robot manipulation data for task planning, object affordance, and end effector trajectories.	Multiple images	Task description
Egocentric Videos			
VSI-Bench Yang et al. (2025a)	Spatial reasoning from sequential observations of indoor environments.	Video	Spatial answer
Perception-Test Patraucean et al. (2023)	Video perception and reasoning over memory, abstraction, physics, and semantics.	Video	Perception answer
OpenEQA Majumdar et al. (2024)	Questions about real world environments using an agent's episodic visual memory.	Video	Environment answer
VidEgoThink Cheng et al. (2024b)	Egocentric understanding across planning, grounding, and reward modeling.	Video	Video reasoning answer
EgoPlan-Bench Chen et al. (2023)	Planning the next step toward a task goal from egocentric task progress.	Video	Next action
Games			
GameQA Tong et al. (2025)	Game state reasoning across video games, including perception and spatial reasoning.	Image	Game state answer
Atari Zhang et al. (2020)	Atari gameplay demonstrations with human actions and simultaneously recorded gaze.	Image	Gameplay action
VideoGameBunny Taesiri & Bezemer (2024)	Understanding video game scenes through captions, questions, and scene attributes.	Image	Game understanding answer
MarioQA Mun et al. (2017)	Event centric, counting, and state questions about Super Mario gameplay.	Video	Gameplay event answer
Spatial Planning			
SpaCE-10 Gong et al. (2025)	Compositional spatial reasoning across multiple scenes, including entity presence.	Multiple images	Spatial description
EmbSpatial-Bench Du et al. (2024)	Object to object and object to scene spatial relationships for embodied tasks.	Image	Spatial relation
Spatial-MM Shiri et al. (2024)	Spatial reasoning about object locations and relations in visual scenes.	Image	Spatial answer
OmniSpatial Jia et al. (2026)	Spatial reasoning across diverse task types and visual scenes.	Image	Spatial answer
3DSRBench Ma et al. (2024)	Reasoning about real world three dimensional locations and spatial relations.	Image	3D spatial answer

E Answer Extraction Logic

To ensure consistent and automated evaluation, we employ a multi-strategy answer extraction logic. Although the prompt instructs models to output only a single letter corresponding to their final choice, outputs can vary. Our programmatic parser is designed to robustly handle these variations.

Extraction strategy. The extraction process follows a prioritized sequence of rules. First, our parser attempts a direct match, checking if the model's entire stripped output is exactly a single, valid option letter (e.g., 'A'). If this primary strategy fails, it deploys a fallback routine that applies a series of regular expression patterns. These patterns are designed to find common answer declarations such as "Answer: A," or "Option A." We note that these regular expressions are occasionally adapted to accommodate the unique, consistent output formats of specific models. If no unambiguous answer can be identified after applying all strategies, the question is skipped and marked unevaluated to maintain the integrity of the evaluation. The complete prompt is available in Appendix K.

F Robustness Analysis of MCQ Format

To ensure the reliability of the multiple-choice format, we conducted extensive ablation studies to quantify the impact of option ordering and rule out random guessing as a confounding factor.

Robustness to Option Ordering. We evaluated 5 representative models across 6 datasets using 5 distinct, seeded random permutations of option orders. As shown in Table 3, the model performance remains highly consistent across permutations, with a standard deviation of approximately 1.9%. This confirms that the rankings reported in AgentVQA are stable artifacts of model capability rather than positional bias.

Table 3: Robustness to Option Ordering: Average accuracy across 5 random permutations of option orders. The low standard deviation indicates high stability.

Shuffle Iteration	**0**	**1**	**2**	**3**	**4**	**Std. Dev**
Average Accuracy	56.0%	54.8%	57.0%	51.7%	54.5%	$\sim 1.9\%$

Refuting random guessing. To verify that model scores reflect genuine engagement with the task, we evaluated constant-selection baselines (e.g., "Select All A", "Select All B") across the benchmark. These strategies yielded accuracies of 28% (A), 27% (B), 23% (C), and 22% (D), averaging **25.0%**, which aligns with the theoretical random baseline for 4-option MCQs. Since even the lowest-performing models in AgentVQA consistently score above this floor, we conclude that the benchmark effectively measures agentic reasoning rather than chance.

G Error Modes

To qualitatively understand model failures, we conduct an in-depth error mode analysis on Gemini 3.1 Pro (Google, 2026). The process involves a three-stage methodology to develop a taxonomy, annotate failures, and verify the results. The resulting distribution of errors is visualized in Figure 8.

Taxonomy development and annotation. First, we prompt the model to output a reasoning chain alongside its final answer for every question. We then manually review a diverse, stratified subset of incorrect predictions to identify and define a comprehensive taxonomy of common failure patterns for each of the five agentic domains. With this taxonomy established, we use Gemini 3.1 Pro for large-scale annotation. For each failure, we provide the model with the visual input, all multiple-choice options, the ground-truth answer, the model's incorrect chosen option, the model's full reasoning chain, and any other relevant metadata. We then prompt it to assign the single most fitting error mode from our predefined taxonomy. The prompt used for assigning error modes is available in Appendix M.

Verification. To ensure the quality of the automated annotations, we perform a final verification step. We manually review a randomly selected subset of the error mode assignments across all domains. This process confirms that the classifications are accurate and consistent with our established definitions, ensuring the reliability of our qualitative analysis.

G.1 Error Mode Definitions

Web Agents. Grounding Error occurs when the model understands the goal but fails to correctly locate or ground the corresponding UI element, often confusing it with a visually similar or nearby one. Logical Breakdown describes a failure to correctly process a sequence of steps, leading to a logically inconsistent action based on the current state. Visual Hallucination is when the model's reasoning relies on a UI element or piece of information that is not present in the visual input. Missing Inference occurs when the model fails to deduce an implicit but necessary action, such as scrolling to reveal an element. Action Mismatch happens when the model's reasoning correctly identifies the right action, but its final output selects a different, incorrect option. Goal Misunderstanding is a fundamental error where the model misinterprets or ignores a key part of the user's instruction.

Egocentric Videos. Motion Misperception occurs when the model fails to correctly interpret a process, motion, or physical interaction as it unfolds over time. Perception Gaps are fundamental failures in seeing, where the model incorrectly reports a basic, static fact like an object's presence or count. Spatial Confusion is a failure to build an accurate 3D mental map from video, resulting in an incorrect understanding of object layout or orientation from a given viewpoint. Incomplete Analysis happens when the model perceives all necessary visual facts but fails to execute the required logical steps to reach the correct conclusion. Task Misinterpretation arises when visual perception is correct, but the model misunderstands the nuance or goal of the prompt.

Robotics. Spatial Confusion occurs when the model fails to correctly interpret the spatial relationships, location, or orientation of objects in 3D space from a 2D image. Object Misrecognition happens when the model misidentifies an object or fails to ground it based on all of its required attributes like color or size. Action-Time Disconnect is a failure to understand the dynamics of a scene, such as predicting the outcome of an action or correctly sequencing a task. Selection Error occurs when the model's reasoning arrives at the correct conclusion, but it selects the wrong option in its final output. COT Failure describes cases where the chain-of-thought reasoning process breaks down, failing to handle specific quantitative values or fine-grained distinctions.

Games. Visual Miscomprehension is a foundational failure to correctly perceive the game's visual and spatial information, such as object locations or relationships. Rule Misunderstanding is a logical error where the model does not correctly apply the game's explicit rules or objectives. Tracking Drift is a failure to mentally simulate and track how the game state changes over time. Reasoning Breakdown is a general failure where the model's internal logic collapses under the task's complexity or becomes confused about the objective, leading to a flawed conclusion. Context Hallucination occurs when the model's reasoning becomes detached from the game state, inventing information that does not exist.

Spatial Planning. Viewpoint Gap occurs when the model identifies objects but fails to construct an accurate 3D map of their relationships, leading to errors in judging depth or distance. Categorization Confusion occurs when perception is visually correct, but the model applies the wrong semantic label or category to an object. Detection Hallucination is a fundamental perceptual failure where the model either misses a present object or "sees" an object that is not there. Logic Collapse is a failure cascade where the model first fails to perceive an object and then tries to compensate by inventing a logical path forward based on hallucinated information. Scope Misinterpretation happens when the model misunderstands the required granularity or intent of the question, particularly for counting or classification tasks.

Table 4: Here we list canonical Vision-Language model agentic benchmarks, organized by domain and their offline/online categorization. We **bold** datasets that are a part of AgentVQA and suffix datasets that have been filtered or filtered and transformed with + or ++ suffixes, respectively.

Domain	Mode	Datasets
Web Agents	Offline	**AitW**$^{++}$ Rawles et al. (2023), **ScreenSpot**$^{++}$ Cheng et al. (2024a), **Monday**$^{++}$ Jang et al. (2025), **Screenspot-Pro**$^{++}$ Li et al. (2025)
	Online	WebArena Zhou et al. (2023), VisualWebArena Koh et al. (2024), TheAgentCompany Xu et al. (2024), AndroidWorld Rawles et al. (2024), VideoWebArena Jang et al. (2024)
Egocentric Videos	Offline	**Perception-Test**$^{+}$ Patraucean et al. (2023), **VSI-Bench**$^{+}$ Yang et al. (2025a), **VidEgoThink** Cheng et al. (2024b), Ego-Exo4D Grauman et al. (2024), **OpenEQA** Majumdar et al. (2024), **EgoPlan-Bench** Chen et al. (2023)
	Online	EgoThink Cheng et al. (2024c)
Robotics	Offline	**RoboRefit**$^{++}$ Lu et al. (2023), **ERQA** Team et al. (2025), **Robo2VLM**$^{+}$ Chen et al. (2025), HoloAssist Wang et al. (2023), X-Embodiment Vuong et al. (2023), **ManipBench**$^{+}$ Zhao et al. (2025), **ShareRobot**$^{+}$ Ji et al. (2025)
	Online	CALVIN Mees et al. (2022), VIMA-Bench Jiang et al. (2022), BEHAVIOR-1K Li et al. (2023b), RoboCasa Nasiriany et al. (2024), EmbodiedBench Yang et al. (2025c)
Games	Offline	**Atari**$^{++}$ Zhang et al. (2020), **GameQA**$^{+}$ Tong et al. (2025), BASALT Milani et al. (2023), **MarioQA**$^{+}$ Mun et al. (2017), **VideoGameBunny**$^{+}$ Taesiri & Bezemer (2024)
	Online	Crafter Hafner (2021), MineDojo Fan et al. (2022), VideoGameBench Zhang et al. (2025), BALROG Paglieri et al. (2024)
Spatial Planning	Offline	**EmbSpatial-Bench**$^{+}$ Du et al. (2024), **SpaCE-10**$^{+}$ Gong et al. (2025), **Spatial-MM**$^{+}$ Shiri et al. (2024), **OmniSpatial**$^{+}$ Jia et al. (2026), **3DSRBench** Ma et al. (2024)
	Online	HabitatChallenge Savva et al. (2019), EXCALIBUR Zhu et al. (2023)

H Limitations of Open-Ended Grounding Metrics

In this section, we visualize why standard open-ended evaluation metrics (Bounding Box IoU and Distance Thresholds) introduce significant noise into agentic evaluation, and how AgentVQA's MCQ format resolves this ambiguity.

As illustrated in Figure 4, open-ended metrics often force a trade-off: loose thresholds or large bounding boxes allow false positives on irregular shapes (like the scissors), while strict thresholds punish valid actions on large elements (like the mouse). AgentVQA circumvents this by transforming the continuous coordinate space into a discrete decision problem, where distractors are programmatically ensured to be incorrect (e.g., outside the ground truth bounds) yet plausible enough to test precision.

I Implications and Suggestions for Practitioners

AgentVQA provides unique diagnostic data unavailable in general VQA benchmarks. Based on the granular failure modes identified in our experiments, we offer the following concrete recommendations for future agentic model design and training:

- **Accelerate Model Iteration:** AgentVQA provides a scalable testbed for cross-domain performance. Evaluating complex agentic questions with a reasoning model judge (generating around 2,000 tokens per decision) requires millions of output tokens for a single pass. AgentVQA mitigates these massive computational costs by enabling rapid, deterministic scoring for faster development cycles.
- **Bridge the Semantic Gap:** Our analysis reveals a semantic-spatial disconnect where models know what to do but fail to locate where to do it. For instance, models often correctly identify the next step in Web Agents but fail the precise coordinate grounding (78% error rate). This indicates that scaling model size is insufficient, and researchers must prioritize specialized grounding pre-training to link high-level intent with low-level execution.
- **Adopt Adaptive Inference:** Since reasoning models often underperform on perception tasks, we recommend adaptive architectures that decouple fast, robust perception modules (for spatial awareness) from expensive Chain-of-Thought (for planning) to prevent "Thinking Loops."
- **Implement Domain-Specific Curricula:** The lack of correlation between general VQA and agentic performance suggests generic tuning is inadequate. Our granular error profiles

suggest a move toward domain-specific curriculum learning, targeting 3D spatial data for Robotics versus causal reasoning for Games.

- **Deploy Offline Safety Gates:** We propose using AgentVQA as a standardized diagnostic filter to detect critical reliability risks such as hallucinations and recursive loops before deploying agents into high-stakes, open-ended environments.

J Analysis of "Thinking Loops" as a Failure Mode

During our analysis of different reasoning models, we identified some models like GLM-4.1V-Thinking getting stuck in a repetitive cycle of reasoning without making any forward progress. For instance, when analyzing one model's reasoning process, we observed it getting stuck in the loop shown below:

```
<think> The person in the video is seen placing a small triangle on the
    table. The next shape that the person places is a large circle. The
    person then places another triangle next to the circle. The person
    then places another triangle next to the circle and the triangle. The
     person then places another triangle next to the circle and the
    triangle. The person then places another triangle next to the circle
    and the triangle........
```

Error mode annotation. We use Gemini 3.1 Pro to categorize model failures into predefined error modes (Section G). All automated assignments were manually verified on a random subset.

K Evaluation Prompt

We use a standard, simple prompt structure that provides the model with the question, any relevant context such as action history, and concludes by instructing it to output only the single letter corresponding to its final choice. The sample prompt structure is mentioned below (Action History would be removed in case of non-trajectory based questions):

Evaluation Prompt Template

This is the standard template used for model evaluation. Placeholders like {episode_goal} are replaced with the specific data for each question.

Goal: {episode_goal}

Action History:
{action_history if action_history.strip() else "No previous actions"}

Question: {question}

Options:
{options_text}

Based on the provided screenshot, goal, action history, and question, select the single best option from the list above.
IMPORTANT: Your response must be EXACTLY one character (A, B, C, or D) with no other text, explanation, or formatting.

Answer:

L HARD NEGATIVE GENERATION PROMPT

This is the primary prompt template used to generate hard negative distractors for datasets like AITW; we use a similar but tailored schema for other datasets requiring transformation.

Context:

- The UI screenshot has dimensions {`width`}x{`height`} pixels.
- The image provided has a green dot marking the location of the correct action if it is a tap.
- The correct action is: **{`correct_action_string`}**

Goal: {`goal`}
Previous Actions (History):
{`history if history else "This is the first step."`}

Your Task:
Generate **three** distinct and plausible but **incorrect** distractor actions. These distractors should be designed to confuse a tester. Use a mix of the following strategies:

1. **Near-Miss:** An action of the same type as the correct one but slightly off (e.g., tapping right next to the correct button).
2. **Semantic-Confusion:** An action on a different but visually or functionally similar element (e.g., tapping 'blacktooth Settings' when 'Wi-Fi Settings' is correct).

Supported Action Formats (Use these formats EXACTLY):

- `Tap: [x, y]` (where x and y are integer pixel coordinates)
- `Swipe: DIRECTION` (where DIRECTION is one of 'Up', 'Down', 'Left', 'Right')
- `Type: 'text to type'` (where 'text to type' is a plausible but incorrect string)
- `Button: ACTION` (where ACTION is one of 'Press Back', 'Press Home', 'Press Enter')

Important Rules:

- Do **NOT** generate the correct answer ({`correct_action_string`}).
- Ensure all tap coordinates are within the image bounds (width: {`width`}, height: {`height`}).
- For "Tap" distractors, analyze the image to choose locations that are genuinely incorrect but tempting.
- Generate a diverse set of three distractors.
- You also need to ensure that for near miss in tap the point is actually a negative.

Respond **ONLY** with a valid JSON object in the following format:

```
{"distractors": ["ACTION_TYPE: value", "ACTION_TYPE: value",
"ACTION_TYPE: value"]}
```

M ERROR MODE ANNOTATION PROMPT

The following prompt is used to have Gemini 3.1 Pro assign a specific error mode to a model's incorrect prediction; the prompt provides full context about the question and the model's failure, along with a detailed taxonomy of all possible error modes and their definitions.

You are an expert AI agent evaluator specializing in embodied robotics and robotic reasoning tasks. Your task is to analyze why a vision-language model failed on a specific question and categorize the failure.

M.1 CONTEXT

QUESTION**:** {`question`}

QUESTION TYPE: {`question_type`}

NUMBER OF IMAGES: {`num_images`}

DATASET: {`dataset_name`}

M.2 VISUAL ANALYSIS

Some description about the expected image scene.

M.3 FAILURE DETAILS

MODEL'S PREDICTION: {`model_prediction`}
CORRECT ANSWER: {`correct_answer`}
MODEL'S REASONING: {`reasoning (if available)`}

TASK

Analyze the model's failure in this embodied robotics reasoning task. Consider whether the model correctly understood the spatial relationships, identified objects and their attributes, predicted action outcomes, handled quantitative aspects, comprehended the instruction, and mapped its reasoning to the correct output. Focus on identifying the root cause using the specialized robotics error modes below.

ERROR MODES

{*A detailed list of all error modes and their definitions is provided here.*}

Respond with a JSON object containing the 'error_mode' and a brief 'justification' for your choice. If you think that none of the error_modes apply then write NaN in error mode and justify in justification and also propose some new error mode in the justification and explain how this one aligns well with this sample as compared to others. It is not always necessary that an error mode applies so you can output NaN along with the justification and closest error mode from list and your idea of error mode for this.