🤖 Open CaptchaWorld: A Comprehensive Web-based Platform for Testing and Benchmarking Multimodal LLM Agents

Yaxin Luo[*], **Zhaoyi Li**[*], **Jiacheng Liu, Jiacheng Cui, Xiaohan Zhao, Zhiqiang Shen**[†]

[1]VILA Lab, MBZUAI   [2]MetaAgentX

[*]Equal Contribution   [†]Corresponding Author

Code & Data: Open CaptchaWorld

Abstract

CAPTCHAs have been a critical bottleneck for deploying web agents in real-world applications, often blocking them from completing end-to-end automation tasks. While modern multimodal LLM agents have demonstrated impressive performance in static perception tasks, their ability to handle interactive, multi-step reasoning challenges like CAPTCHAs is largely untested. To address this gap, we introduce **Open CaptchaWorld** 🤖 , the first web-based benchmark and platform specifically designed to evaluate the visual reasoning and interaction capabilities of MLLM-powered agents through diverse and dynamic CAPTCHA puzzles. Our benchmark spans 20 modern CAPTCHA types, totaling 225 CAPTCHAs, annotated with a new metric we propose: *CAPTCHA Reasoning Depth*, which quantifies the number of cognitive and motor steps required to solve each puzzle. Experimental results show that humans consistently achieve near-perfect scores, state-of-the-art MLLM agents struggle significantly, with success rates at most **40.0%** by Browser-Use Openai-o3, far below human-level performance, **93.3%**. This highlights Open CaptchaWorld as a vital benchmark for diagnosing the limits of current multimodal agents and guiding the development of more robust multimodal reasoning systems.

1 Introduction

Multimodal agents powered by large language models (LLMs) [40, 11, 25, 4, 3, 27, 7] are rapidly advancing toward real-world deployment, with the promise of automating tasks such as form filling, navigation, shopping and other interactions on websites. However, one major roadblock remains: CAPTCHAs. These human verification puzzles, designed to prevent bots from abusing web services, frequently prevent agents from completing real tasks, especially on high-value sites like e-commerce platforms or login pages. For agent-based systems to be truly deployable in the wild, solving CAPTCHAs autonomously must become a core capability.

Recent Multimodal LLMs (MLLMs) such as Openai-o3 [27], Claude-3.7-Sonnet [2], and Gemini2.5-Pro [7] have demonstrated strong capabilities across a range of visual-language tasks, including object grounding [31, 46, 38], VQA [9, 12, 22, 35], and document analysis [23, 13, 50]. They can observe screenshots, interpret UI elements, and issue text or click-based commands. Yet these models are usually tested in static, one-shot benchmarks, lacking the multi-step, tool-using, and interaction-heavy dynamics found in CAPTCHA tasks. As a result, we still lack a reliable assessment of whether these models can reason and act like humans in complex, vision-guided interactions.
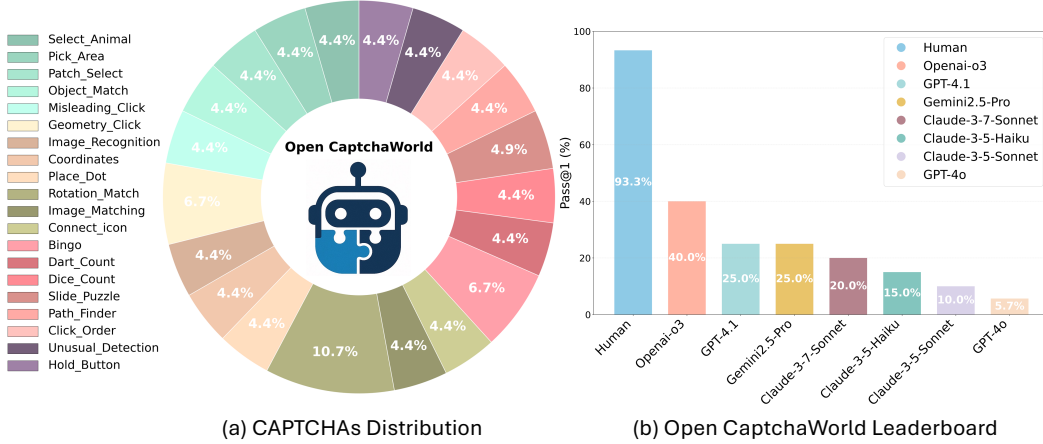
Preprint.

Figure 1: **Open CaptchaWorld data distribution and MLLMs performance plot.**

Despite the explosion of agent benchmarks, most systematically filter out CAPTCHAs. Visu-alWebArena [15] and AgentBench [19] simulate realistic environments but discard pages with CAPTCHAs [43]. Traditional CAPTCHA-solving work (e.g., Deep-CAPTCHA [26], Breaking reCAPTCHAv2 [30]) treats them as static perception tasks solvable by CNNs or object detectors, ignoring the sequential planning and interface state dynamics. This leaves a crucial evaluation gap: no benchmark tests whether MLLM agents can handle CAPTCHAs in a closed-loop, interactive setting that mimics real-world browsing.

To close this gap, we introduce **Open CaptchaWorld**, a web-based benchmark designed to assess whether agents can autonomously solve modern CAPTCHAs through perception, reasoning, and multi-step interaction. Our benchmark includes drag-based, sequence-click, slider alignment, and counting-based puzzles, all designed to be intuitive for humans but challenging for current agents. Unlike prior work that filters CAPTCHAs out, we embrace them as essential obstacles for agent robustness and autonomy.Our benchmark consists of 20 diverse CAPTCHA types, the number of each type will be continuously increasing, and a novel metric called **CAPTCHA Reasoning Depth**, which quantifies how many cognitive and motor steps are needed to solve the task. Despite its modest size, Open CaptchaWorld represents a highly challenging and realistic benchmark for agent-based multimodal reasoning, owing to its interactive nature, step-by-step decision requirements, and high variance in visual-cognitive complexity. All puzzles are tested in a real browser loop, where agents must perceive screenshots and issue clicks or key actions until the task is complete.

We evaluate a broad spectrum of the most advanced MLLM models equipped with browser-use tools [24], including Openai-o3, Claude-3.7-Sonnet, Gemini2.5-Pro, and GPT-4.1 etc, find that success rates vary widely by puzzle type and depth. Notably, even top-performing agents lag behind humans by **-53.3%**. Moreover, the benchmark is explicitly designed to test generalization and reasoning depth, not memorization from massive data. As our evaluations show, state-of-the-art agents perform far below human levels **Our main contributions are as follows:** (1) We propose **Open CaptchaWorld**, the first open-source, large-scale, and long-term maintaining CAPTCHA benchmark for evaluating interactive multimodal agents using MLLMs. (2) We introduce **CAPTCHA Reasoning Depth**, a task-agnostic complexity measure capturing the multi-step reasoning burden of visual interaction puzzles. (3) We build a real web-based testing platform[1] and systematically evaluate state-of-the-art models in zero-shot settings, revealing large performance gaps compared to humans. (4) We provide insights into agent failure cases such as overthinking, over-segmentation and interface misunderstanding.

2 Related Work

The evolution of multimodal LLMs (MLLMs) such as Openai-o3 [27], Gemini2.5-Pro [7], and Deepseek-V3 [41] has been driven by increasingly diverse benchmarks [1, 16, 18, 52, 4, 3], ranging from math [21], visual QA [10, 12, 22], to OCR-based reasoning [35]. To assess these models comprehensively, benchmarks like MMBench [20], MME [6], MMMU [48], and MM-Vet [47]
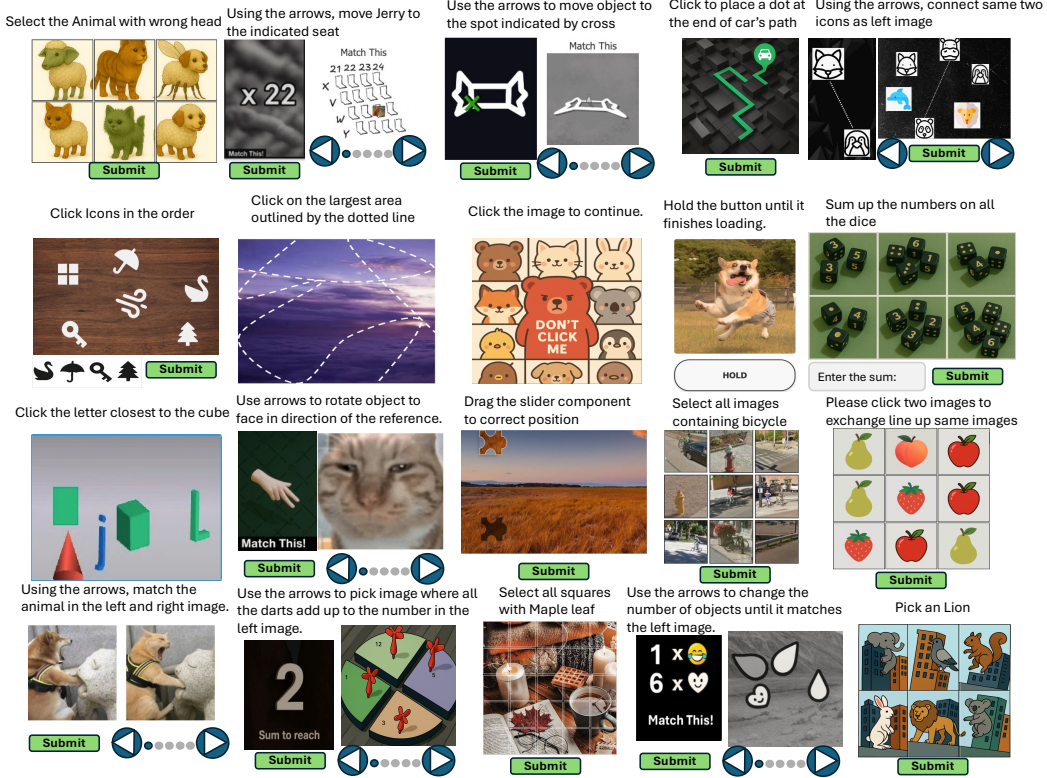
---

[1]https://huggingface.co/spaces/OpenCaptchaWorld/platform.

Figure 2: **Examples from Open CaptchaWorld.**

evaluate a wide range of MLLM capabilities. However, most assume a static, single-turn setup [45], limiting their ability to test dynamic, real-world interaction.

To overcome this, recent work has explored LLM and MLLM agents operating in interactive environments [29, 37, 33], often with external tool use [49, 5, 8, 17, 32] and multi-step decision-making [44, 39, 34]. Benchmarks like SWE-bench [14] test an agent's ability to debug and patch codebases, while WebArena [51] and its multimodal extension VisualWebArena [15] require agents to interpret text and images to complete web-based goals. AgentBench [19] aggregates tasks across diverse domains, and ToolBench [15] isolates tool-use challenges.

However, CAPTCHAs remain underexplored in this agentic paradigm. Existing solutions [26, 30] treat CAPTCHA solving as static vision tasks, ignoring interactive challenges like UI state tracking, fine-grained control, and sequential decision-making. In contrast, modern LLM agents integrate perception, reasoning, and action [44, 34], making them suitable for solving complex CAPTCHA puzzles in dynamic environments. Despite progress in multi-turn reasoning benchmarks, no open-source efforts target CAPTCHA solving in the way AgentBench [19] or VisualWebArena [15] test broader interactions. Our work fills this gap by introducing a web-based CAPTCHA benchmark where MLLM agents must perceive, plan, and act over multiple steps, providing a realistic testbed for evaluating agent robustness beyond static classification.

3 Open CaptchaWorld

Open CaptchaWorld is a carefully curated benchmark designed to evaluate multi-step, interactive visual reasoning CAPTCHAs that are hard for models but easy for humans to solve. Inspired by commercial CAPTCHA systems like Google's reCAPTCHA, Arkose Labs' Arkose MatchKey. We systematically design and annotate images to construct Open CaptchaWorld web-based benchmark for Multimodal Agents. All images are either drawn by human designers or generated by GPT-4o [28].

3.1 Open CaptchaWorld serves as a complement to Web Agent's benchmarks

With the progress of Agent's development, the web agents will finally be deployed in real-world applications to automatically finish tasks on websites. However, we notice that previous research

usually ignores websites that contain CAPTCHAs, because tasks involving websites with CAPTCHA prevent agents from completing the task. However, those websites are usually more commercial and popular websites, which contain more real-life, day-to-day tasks. Besides web Agents, the existing benchmarks usually discard web pages that contain a CAPTCHA system when they construct their benchmarks [42]. However, in order to deploy web agents in the real world, the CAPTCHAs can not be easily ignored and skipped; we need to develop solutions for web agents to tackle this challenge.

To address this overlooked yet crucial challenge, Open CaptchaWorld is introduced as a dedicated benchmark that explicitly targets web environments containing CAPTCHAs. Unlike prior datasets that filter out these interaction barriers, Open CaptchaWorld embraces them as necessary components for evaluating the readiness of web agents in real-world deployments. CAPTCHAs are not edge cases, which are commonly encountered in high-value, security-sensitive websites such as ticketing platforms, e-commerce portals, and account login flows. Bypassing them in evaluation leads to a misleading sense of agent competence. Open CaptchaWorld systematically curates a diverse set of CAPTCHA puzzles, spanning image-based selection, drag-and-drop mechanics, jigsaw alignment, and object counting. These scenarios go beyond static perception, which requires agents to combine multimodal understanding, memory across steps, and dynamic interaction with on-page elements. As such, this benchmark shifts the focus from single-turn prediction to interactive problem-solving, a key trait for practical autonomy.

3.2 CAPTCHA Reasoning Depth

To better characterize cognitive difficulty of puzzles in *Open CaptchaWorld*, we introduce a new metric called "*CAPTCHA Reasoning Depth*", which quantifies the number of reasoning and interaction steps a human must perform to solve a given CAPTCHA. Unlike traditional classifications that group puzzles by type (e.g., image selection, jigsaw, or drag tasks), reasoning depth offers a task-agnostic measure of complexity that aligns more closely with the multi-step nature of agent reasoning. We define CAPTCHA Reasoning Depth as the minimal number of atomic reasoning or decision-making steps required by a human or a model to arrive at a correct solution, where each step involves interpreting visual content, planning a subgoal, or executing a discrete interaction (e.g., a drag, click, or alignment operation). Formally, let a CAPTCHA be defined as a task T requiring a sequence of operations. We define the CAPTCHA Reasoning Depth $D(T)$ as:

$$D(T) = \sum_{i=1}^{N} \mathbb{I}[s_i \in \mathcal{S}_T] \tag{1}$$

where $\mathcal{S}_T$ is the set of atomic steps needed to solve T, s_i is an atomic reasoning or interaction step from a predefined checklist $\mathcal{C}$ (see Table 3), and $\mathbb{I}[\cdot]$ is the indicator function. Each s_i contributes 1 unit of depth if the step is observed during the solution process. The checklist $\mathcal{C}$ includes categories such as visual perception, cognitive planning, motor control, and state monitoring.

For instance, a puzzle that asks the user to "click on the fox" typically requires two steps: first, identify the target object among distractors, and second, perform the click. In contrast, a drag-based jigsaw CAPTCHA may require identifying multiple part alignments, sequencing them appropriately, and dragging each piece to its correct location, leading to a reasoning depth depending on puzzle layout and ambiguity.

To measure this across the benchmark, we conducted a human annotation study where participants were asked to solve a sample of puzzles while verbally annotating each reasoning step they performed. Annotators were instructed to decompose their process into a sequence of atomic steps and actions. And we construct heuristic rules to guidance the annotators to make their responses consistent, the rules in Table 3. We then recorded the number of steps and averaged across annotators to estimate the reasoning depth per puzzle. We also computed inter-annotator agreement and variance to assess consistency across participants. To better compare the reasoning depth difference for human and LLM agents to solve the CAPTCHAs, we also prompt Openai-o3 [27] and Gemini2.5-Pro [7] with the previous heuristic rules to estimate the reasoning depth of each type of CAPTCHAs, the detailed prompt is in Fig. 10. For humans' estimation of reasoning depth to each CAPTCHA Fig. 3 shows the distribution. Puzzles span a wide range of depths, illustrating the diverse difficulty levels for humans. Across the dataset, we observe high structural diversity: the average reasoning depth per task is 2.94 with a standard deviation of 0.92. This confirms the benchmark covers a wide range of cognitive
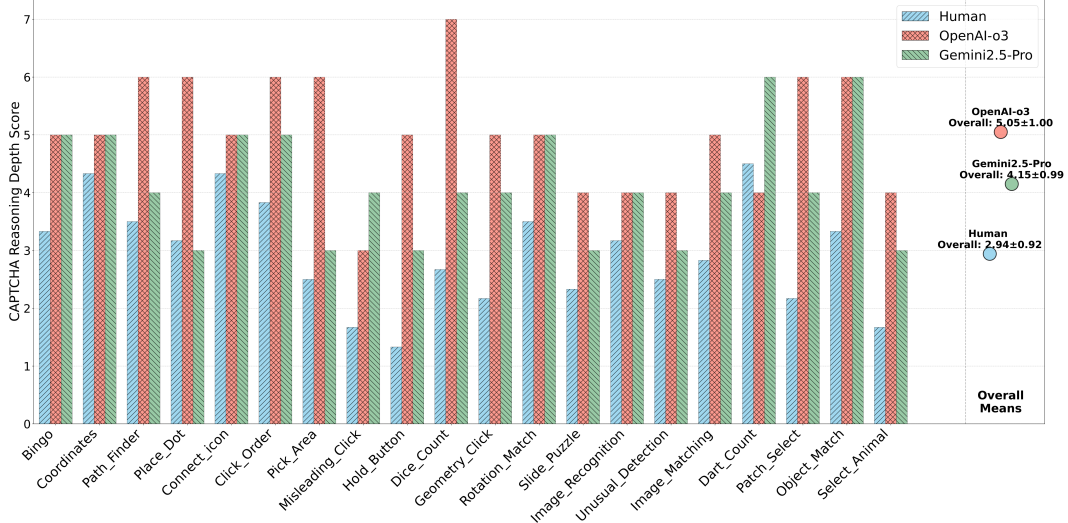
Figure 3: CAPTCHA Reasoning Depth Estimation by Human Annotators and Most Advanced Reasoning Models.

difficulty levels. Furthermore, each CAPTCHA type is instantiated with at least 10 diverse variants, manually crafted or generated with variation in spatial layout, icon types, or interaction mode.

Different Reasoning Depth Estimate Behavior Between Human and Models. To better understand why MLLM models and humans provide different reasoning depth estimations shown in Fig. 3, we compare their thinking processes when analyzing the same CAPTCHA. Fig. 4 illustrates an example to this difference. For example, in a sequence-matching CAPTCHA, the human annotator simply identifies the icon order from reference image, searches for them in main panel, clicks each in sequence, and submits the answer, resulting in a depth score of 3. Humans focuses only on key goal-directed actions, compressing low-level perception and memory usage into intuitive, seamless behavior. In contrast, the Openai-o3 model oversegments the process. It lists granular steps such as recognizing each icon, memorizing their order, executing each click separately, and monitoring interface feedback after every action. This leads the model to assign a higher reasoning depth. The model treats each sub-action (e.g., "confirm progress" or "hold cue in memory") as a distinct reasoning unit, even when humans would consider them implicit or automatic.

This example reinforces a broader pattern we observe across the benchmark: models tend to overthink by breaking tasks into fine-grained, literal steps, while humans rely on holistic understanding and prior experience to simplify their reasoning. Humans can skip over obvious or familiar operations and focus on solving the puzzle efficiently. Another key difference is memory. Humans can leverage lifelong experience with similar puzzles and apply learned patterns without deliberation. In contrast, models reset their context at beginning of each conversation and cannot reuse prior exposure unless explicitly prompted. They also lack common-sense filtering, treating all instructions and UI elements as equally important, which further inflates their reasoning depth estimates. This discrepancy highlights a core challenge in building effective agent systems: achieving human-like efficiency, intuition, and abstraction in multi-step reasoning. A robust benchmark must capture this behavioral gap.

3.3 Dataset Curation

As existing CAPTCHAs are for commercial use and not open-sourced, we can not collect them online. Hence, we develop a data curation pipeline to construct the first open-sourced CAPTCHA dataset. The images in our dataset are either generated by GPT-4o [28] or from human designers. To make data reliable, we use human annotators to create groundtruth and instructions. Fig. 5 demonstrates the pipeline to construct our dataset. We first brainstorm, search, and collect twenty CAPTCHA types. Then, for each type, the images are either generated from GPT-4o or designed by human artists. After we have all the images we need, we will design a modern CAPTCHA tasks for each type which will need a multi-step, long horizon, and interactive actions (e.g., click, drag mouse cursor) task solving ability, notice that we do not test model's broad knowledge, so each CAPTCHA is actually could be solved by humans easily but hard for LLM Agents. Then, in step three, each type of CAPTCHA will
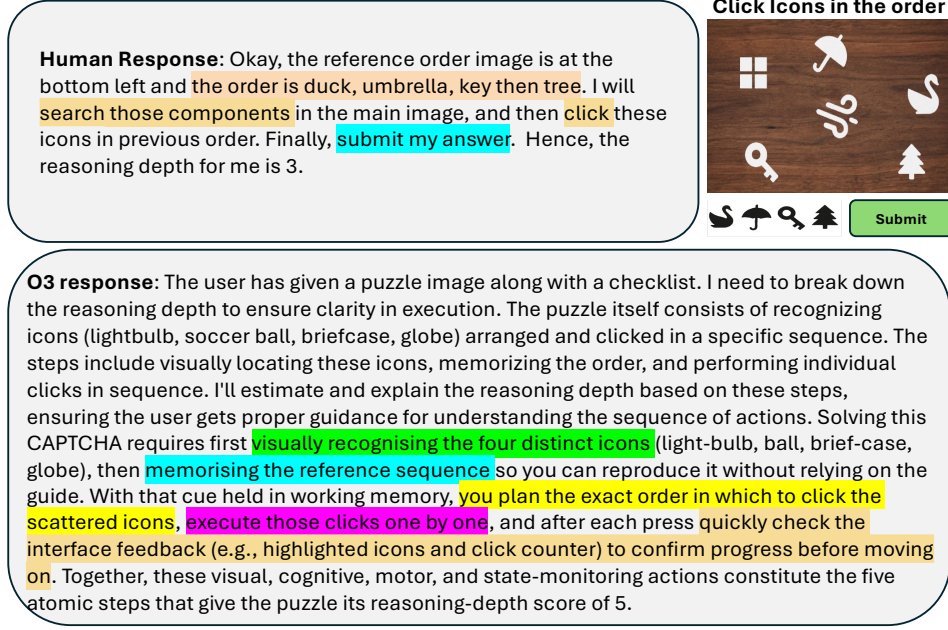
5

Human Response: Okay, the reference order image is at the bottom left and the order is duck, umbrella, key then tree. I will search those components in the main image, and then click these icons in previous order. Finally, submit my answer. Hence, the reasoning depth for me is 3.

Click Icons in the order

O3 response: The user has given a puzzle image along with a checklist. I need to break down the reasoning depth to ensure clarity in execution. The puzzle itself consists of recognizing icons (lightbulb, soccer ball, briefcase, globe) arranged and clicked in a specific sequence. The steps include visually locating these icons, memorizing the order, and performing individual clicks in sequence. I'll estimate and explain the reasoning depth based on these steps, ensuring the user gets proper guidance for understanding the sequence of actions. Solving this CAPTCHA requires first visually recognising the four distinct icons (light-bulb, ball, brief-case, globe), then memorising the reference sequence so you can reproduce it without relying on the guide. With that cue held in working memory, you plan the exact order in which to click the scattered icons, execute those clicks one by one, and after each press quickly check the interface feedback (e.g., highlighted icons and click counter) to confirm progress before moving on. Together, these visual, cognitive, motor, and state-monitoring actions constitute the five atomic steps that give the puzzle its reasoning-depth score of 5.

Figure 4: Thinking Process Comparison When Estimating CAPTCHA Reasoning Depth between human and Openai-o3 model.
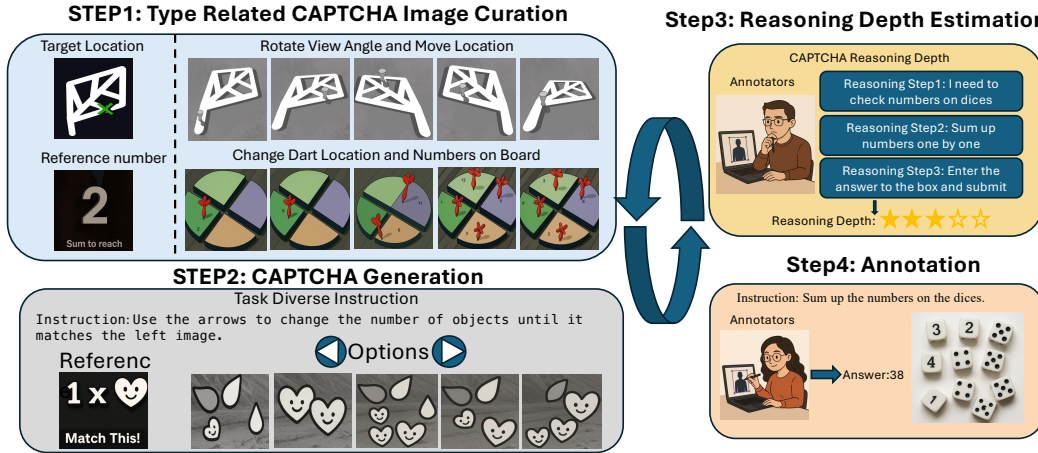

Figure 5: Open CaptchaWorld Date Curation Pipeline. **Step 1**: Curate diverse visual variations for each CAPTCHA type by modifying object positions, angles, and contextual cues. **Step 2**: Generate interactive tasks with human- or GPT-generated instructions tied to each image. **Step 3**: Estimate the CAPTCHA Reasoning Depth by decomposing the human solving process into atomic reasoning steps. **Step 4**: Annotate final answers and instructions to ensure high-quality, human-solvable groundtruth for model evaluation.

be marked with our previously proposed *CAPTCHA Reasoning Depth* metrics by human annotators, this metrics and annotations can help us understand the different behaviors and misalignment of LLM Agents and humans when compared with their attempts to solve the CAPTCHAs. After all, the final ground truth solutions of CAPTCHAs will be annotated by annotators to make sure the ground truth is reliable, as humans can perform a 93.3% success rate in such a CAPTCHA environment, while LLM Agents are still far behind human performance. In addition, we show 20 examples from our Open CaptchaWorld in Fig. 2, covering all the types in dataset.

3.4 Multimodal Agents solve CAPTCHA

After curating the dataset and deploying our benchmark platform, we model the CAPTCHA-solving process of an agent as a finite-horizon partially observable Markov decision process (POMDP) [36],

defined by the tuple:

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{Z}, R, \gamma) \tag{2}$$

where $\mathcal{S}$ is the latent environment state (e.g., CAPTCHA interface configuration), $\mathcal{A}$ is the action space (e.g., clicks, drags), $\mathcal{O}$ is the observation space (e.g., screenshots), $\mathcal{T}(s'|s, a)$ is the state transition probability, $\mathcal{Z}(o|s)$ is the observation function, $R(s, a)$ is the reward (success or failure), and γ is the discount factor (we set to 1 as we model CAPTCHA types equally) .

At each time step t, the agent receives an observation $o_t \in \mathcal{O}$ (e.g., screenshot), infers a belief state b_t, and selects an action $a_t \in \mathcal{A}$. The environment transitions to a new state s_{t+1} and produces a new observation o_{t+1}. The agent aims to maximize the expected cumulative reward over the episode:

$$\mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t) \right] \tag{3}$$

This expression reflects the agent's strategy of selecting actions that lead to successful CAPTCHA completion, balancing immediate and future rewards over the episode.

4 Empirical Analysis

We systematically evaluate both base multimodal models and agent-based reasoning approaches on Open CaptchaWorld benchmark. To ensure fair comparisons, we adopt a unified experimental setup with consistent prompting strategies and evaluation metrics applied across models and methods. In Section 4.1, we describe our evaluation protocol and implementation of Browser Use agents [24] equipped with different MLLM backbones. Section 4.2 presents the success rates of various models across all CAPTCHA types, highlighting the overall performance gap between humans and current agents. We then dive deeper in Section 4.3, conducting a fine-grained case study of success and failure patterns, categorized by task type and reasoning demand. Together, these analyses shed light on current limitations of multimodal agents and offer practical implications for future model design.

Table 1: Performance of different MLLM backbones within the Browser Use baseline agent on Open CaptchaWorld. Darker "▇" indicates higher success rate@1 and darker "▇" indicates higher cost.

Solver Type	MLLM Backbone	Pass@1 (%)	Cost ($)
Human	–	93.30	-
Browser Use Agents	GPT-4o	5.7	25.8
	GPT-4.1	25.0	16.7
	Claude-3.7-Sonnet	20.0	18.7
	Gemini2.5-Pro	25.0	18.1
	Openai-o3	40.0	66.4
	Claude-3.5-Haiku	15.0	9.3
	Claude-3.5-Sonnet	10.0	21.9
	Openai-o1	5.0	94.6
	DeepSeek-V3	20.0	7.3

4.1 Experimental Setup

4.2 Success Rate of Multimodal Agents on Open CaptchaWorld

We evaluate our benchmark in a zero-shot setting using 20 types of modern CAPTCHA puzzles. To better reflect real-world interaction needs and test powerful MLLM agents, we exclude traditional CAPTCHA formats such as distorted text recognition or static image classification as they can be even solved by simple detection and classification models. All experiments are run in a web-based

testing environment, where agents can perform multi-step actions like clicking, dragging, or typing. The CAPTCHAs are shown in a type-by-type sequence without repetition, ensuring that agents go through all puzzle types exactly once. We implement a Browser-Use Agent [24] system powered by different multimodal language models (MLLMs), including GPT-4o, GPT-4.1 (2025-04-14), Claude-3.7-Sonnet, Claude-3.5-Sonnet, Claude-3.5-Haiku, Gemini2.5-Pro, DeepSeek-V3, and Openai-o3 (2025-04-16). These agents operate in a closed-loop setup: they receive screenshots of browser, reason about task, and issue actions step-by-step until they click final submit button. Moreover, the prompt we used to test Multimodal Agents is in Fig. 11.
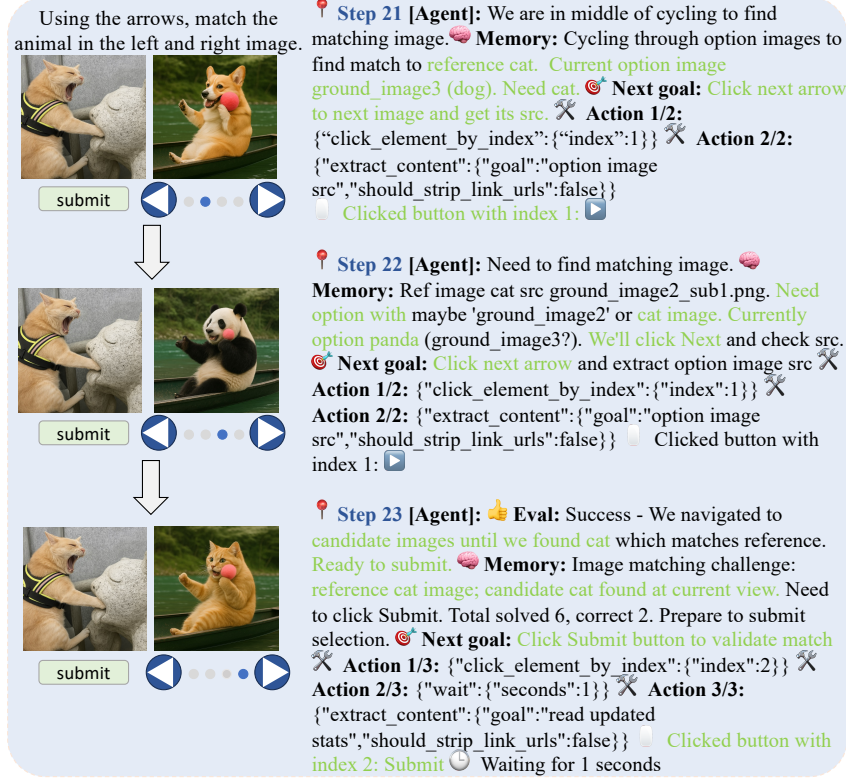

Figure 6: Step-by-step reasoning process of Openai-o3 in successfully solving Image Matching.

Table 1 presents the pass@1 success rate of various most advanced MLLM-powered browser-use agents on the Open CaptchaWorld benchmark. While human participants achieve an average success rate of 93.3%, all current models fall significantly short. The strongest performer, Openai-o3, reaches 40.0%, followed by GPT-4.1 and Gemini2.5-Pro at 25.0%. Other models, including Claude and GPT-4o variants, perform between 5.0% and 20.0%, with several showing near-random behavior on more complex tasks.

In addition to performance, we also report the cost per evaluation episode in USD$, as shown in Table 1 and Fig 7. While Openai-o3 demonstrates the best success rate among agents, it
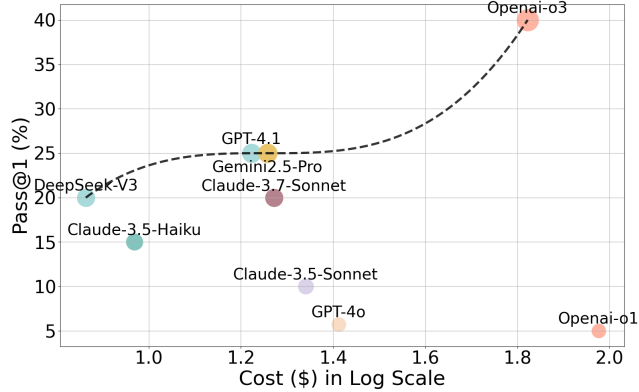

Figure 7: **Cost-performance trade-off among browser-use agents.** Each point represents a model, plotted by its evaluation cost (in log scale) and pass@1 success rate on Open CaptchaWorld. Openai-o3 achieves the highest success rate but incurs substantial cost, while models like Gemini2.5-Pro offer more favorable cost-effectiveness.

also incurs a high cost of $66.4 per full CAPTCHA sequence, and GPT-4o and Claude-3.7-Sonnet show much lower performance at a moderate cost range. Notably, Openai-o1 yields the lowest
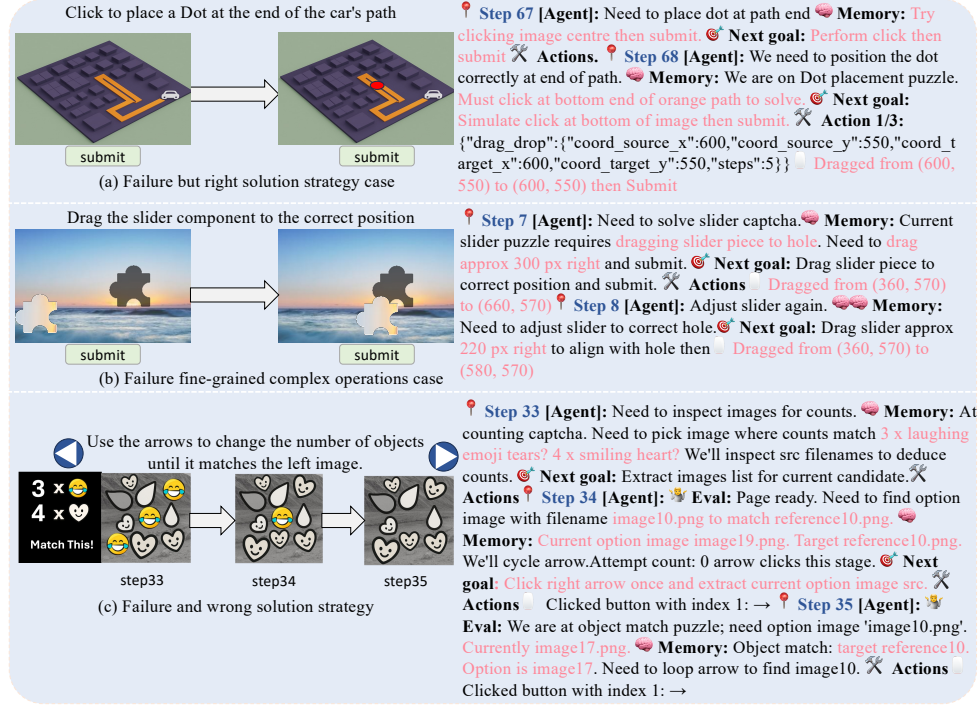
Figure 8: **Representative Failure of Openai-o3 Across Challenging CAPTCHA Types.** (a) Failure case with correct strategy but limited visual perception. (b) Failure case due to complex operational execution. (c) Failure case caused by misguided solution strategy based on irrelevant cues.

success rate (5.0%) while being the most expensive ($94.6), making it the least cost-effective option. In contrast, models like DeepSeek-V3 and Claude-3.5-Haiku offer a more favorable balance of cost and performance, albeit at a relatively low accuracy.

These results highlight that model choice involves not only accuracy tradeoffs but also budget considerations, especially when deploying CAPTCHA-solving agents at scale. Cost-effective but robust agents remain an open challenge. Overall, the wide variance in both success rates and cost underscores the need for more efficient, reasoning-aligned MLLMs capable of performing real-world multi-step interactions with both accuracy and resource awareness.

4.3 Success and Failure Cases Analysis

As shown in Table 2, most models perform well on CAPTCHA types that rely primarily on basic visual perception, such as Image Recognition, Image Matching, Object Match, and especially Select Animal. Beyond these common types, Openai-o3 also succeeds on more challenging tasks like Dart Count and Rotation Match, which require arithmetic and spatial reasoning. Notably, Claude-3.7-Sonnet and Claude-3.5-Haiku go further by handling Bingo-type CAPTCHAs, with Claude-3.7-Sonnet uniquely excelling at the Hold Button task, indicating a higher level of operational reasoning.

Given its strong overall performance and structured reasoning, we select Openai-o3 as a representative model to analyze across 20 CAPTCHA types, focusing on both successes and failures to assess its visual and cognitive abilities. Openai-o3 consistently solves tasks such as Object Match, Image Recognition, Select Animal, Image Matching, Dart Count, Rotation Match, and Patch Select. These tasks primarily depend on visual perception, object recognition, and basic reasoning, without requiring complex inference or interaction. Fig. 6 shows a successful example of o3 solving an Image Matching CAPTCHA: the model iteratively evaluates the current state, updates its memory, sets a goal, and cycles through candidate images until a match is found and submitted.

To better understand Openai-o3 model's limitations, we categorize its failure cases across challenging CAPTCHA types into three representative patterns, as illustrated in Fig. 8. These include: (a) failures where the model follows a generally correct solution strategy but lacks sufficient visual perception or spatial understanding, for instance, in the Place_Dot task, it assumes the dot should be placed at the

end of the path but repeatedly clicks near the center, missing the actual target; (b) failures involving fine-grained but complex operations, such as in the Slide_Puzzle task, where the model understands the goal but fails to compute and execute the precise alignment needed; and (c) failures resulting from misguided strategies, such as in the Object_Match task, where the model relies on image filenames or HTML text cues rather than visual analysis, leading to fundamentally incorrect solutions.

5   Conclusion

We introduce **Open CaptchaWorld**, the first open-source, web-based CAPTCHA benchmark designed to evaluate the interactive reasoning capabilities of multimodal LLM agents through diverse modern CAPTCHA puzzles. Our benchmark highlights a crucial yet overlooked challenge in deploying real-world agents: the ability to perceive, reason, and act over multi-step tasks in dynamic web environments. By incorporating 20 diverse CAPTCHA types and introducing the CAPTCHA Reasoning Depth metric, we provide a task-agnostic measure of visual-cognitive difficulty. Empirical evaluations reveal a wide gap between human and model performance, with even top agents like Openai-o3 achieving only 40% success rate compared to 93.3% for humans. Through detailed failure case analysis and observations of model overthinking behavior, we uncover fundamental limitations in current agent reasoning. Open CaptchaWorld thus offers a rigorous testbed for diagnosing weaknesses and guiding the development of more robust, human-aligned multimodal agents.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, James Tilsted, Karen Simonyan, João Carreira, Erich Elsen, Matthias Minderer, et al. Flamingo: A visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.

[2] Anthropic. Claude 3.7 Sonnet System Card. `https://www.anthropic.com/claude-3-7-sonnet-system-card`, 2025. Technical report, February 2025.

[3] Chunyuan Chen, Hao Li, Zhengxuan Liu, Yong Wang, Yi Zhou, Hangbo Li, Yue Li, Zhirui Liu, and Furu Wei. Qwen-vl: A versatile vision–language model for perception, localization, and generation. *arXiv preprint arXiv:2308.12966*, 2023.

[4] Zihao Dou, Feng Wang, Lin Zhang, Shidong Liu, Shuai Lu, Luming Ding, Wengang Wang, Bo Wang, Lei Li, and Song Bai. Internvl: Scaling up vision foundation models and aligning for generic vision–language understanding. *arXiv preprint arXiv:2312.14238*, 2023.

[5] Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*, 2025.

[6] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2024.

[7] Google DeepMind. Gemini 2.5 pro: Our most intelligent ai model. `https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/`, 2025. Blog post, March 2025.

[8] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. Tora: A tool-integrated reasoning agent for mathematical problem solving. *arXiv preprint arXiv:2309.17452*, 2023.

[9] Yash Goyal, Tejas Khot, Daniel Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in VQA matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017.

[10] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[11] Yanheng He, Jiahe Jin, Shijie Xia, Jiadi Su, Runze Fan, Haoyang Zou, Xiangkun Hu, and Pengfei Liu. Pc agent: While you sleep, ai works–a cognitive journey into digital world. *arXiv preprint arXiv:2412.17589*, 2024.

[12] Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *arXiv preprint arXiv:1902.09506*, 2019.

[13] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. FUNSD: A dataset for form understanding in noisy scanned documents. *arXiv preprint arXiv:1905.13538*, 2019.

[14] 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*, 2024.

[15] 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.

[16] Junnan Li, Dongxu Li, Caiming Xiong, and H. Hoi, Steven C.˙ Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[17] Xuefeng Li, Haoyang Zou, and Pengfei Liu. Torl: Scaling tool-integrated rl. *arXiv preprint arXiv:2503.23383*, 2025.

[18] Haotian Liu, Simon Jenni, and Jia Deng. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

[19] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, and et al. Agentbench: Evaluating LLMs as agents. *arXiv preprint arXiv:2308.03688*, 2023.

[20] 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 *European conference on computer vision*, pages 216–233. Springer, 2024.

[21] 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, 2024.

[22] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Alexander Schwing. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *CVPR*, 2019.

[23] Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for visual question answering on document images. In *WACV*, 2021.

[24] Magnus Müller and Gregor Žunič. Browser use: Enable ai to control your browser, 2024. URL `https://github.com/browser-use/browser-use`.

[25] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2022.

[26] Zahra Noury and Mahdi Rezaei. Deep-captcha: a deep learning based captcha solver for vulnerability assessment, 2020.

[27] OpenAI. Openai o3 and o4-mini system card. `https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf`, 2025. Technical report, April 2025.

[28] OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, , et al. Gpt-4o system card, 2024.

[29] Siqi Ouyang and Lei Li. Autoplan: Automatic planning of interactive decision-making tasks with large language models. In *Findings of EMNLP*, 2023.

[30] Andreas Plesner, Tobias Vontobel, and Roger Wattenhofer. Breaking recaptchav2. In *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*, page 1047–1056. IEEE, July 2024. doi: 10.1109/compsac61105.2024.00142. URL `http://dx.doi.org/10.1109/COMPSAC61105.2024.00142`.

[31] Bryan A. Plummer, Liwei Wang, Cristina Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30K entities: Collecting region-to-phrase correspondences for richer image–to–sentence models. In *ICCV*, 2015.

[32] Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025.

[33] Nikolai Rozanov and Marek Rei. Stateact: Enhancing llm base agents via self-prompting and state-tracking. *arXiv preprint arXiv:2410.02810*, 2024.

[34] Significant Gravitas. Autogpt: An autonomous gpt-4 experiment. `https://github.com/Significant-Gravitas/AutoGPT`, 2023. GitHub repository.

[35] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read, 2019.

[36] Matthijs TJ Spaan. Partially observable markov decision processes. In *Reinforcement learning: State-of-the-art*, pages 387–414. Springer, 2012.

[37] Haotian Sun, Yuchen Zhuang, Lingkai Kong, Bo Dai, and Chao Zhang. Adaplanner: Adaptive planning from feedback with language models. *arXiv preprint arXiv:2305.16653*, 2023.

[38] Chenyun Wang, Xiaohui Shen, Zhicheng Lin, and Scott Cohen. Phrasecut: Language grounding in images by text-based mask segmentation. *ECCV*, 2020.

[39] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

[40] Junyang Wang, Haiyang Xu, Jiabo Ye, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. Mobile-agent: Autonomous multi-modal mobile device agent with visual perception. *arXiv preprint arXiv:2401.16158*, 2024.

[41] Liang Wei, Jiaxing Zhang, Yue Wang, Meiyu Liu, Zhi Hu, Yiming Wang, Shikun Wang, Ziqi Zhang, Xingtian Dong, and Long Zhou. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[42] Tianci Xue, Weijian Qi, Tianneng Shi, Chan Hee Song, Boyu Gou, Dawn Song, Huan Sun, and Yu Su. An illusion of progress? assessing the current state of web agents. *arXiv preprint arXiv:2504.01382*, 2025.

[43] Tianci Xue, Weijian Qi, Tianneng Shi, Chan Hee Song, Boyu Gou, Dawn Song, Huan Sun, and Yu Su. An illusion of progress? assessing the current state of web agents, 2025.

[44] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[45] 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, 2025.

[46] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling relationships in referring expressions with compositional modular networks. In *CVPR*, 2016.

[47] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

[48] 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, 2024.

[49] Shaokun Zhang, Yi Dong, Jieyu Zhang, Jan Kautz, Bryan Catanzaro, Andrew Tao, Qingyun Wu, Zhiding Yu, and Guilin Liu. Nemotron-research-tool-n1: Tool-using language models with reinforced reasoning. *arXiv preprint arXiv:2505.00024*, 2025.

[50] Xu Zhong, Jianbin Tang, and Antonio Jimeno-Yepes. PubLayNet: Largest dataset ever for document layout analysis. *arXiv preprint arXiv:1908.07836*, 2019.

[51] Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2024.

[52] Damo Zhu, Junyang Chen, Junnan Yang, Weijie Xu, Heyang Zhang, Jianxin Zhang, Yan Zhang, and Jianlong Chen. Minigpt-4: Enhancing vision–language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

Appendix

A   More Examples from Open CaptchaWorld

Here we provide more examples of CAPTCHAs in our Open CaptchaWorld Benchmark, please see Figure 9. Notice that all the images for each CAPTCHA are not repeated.
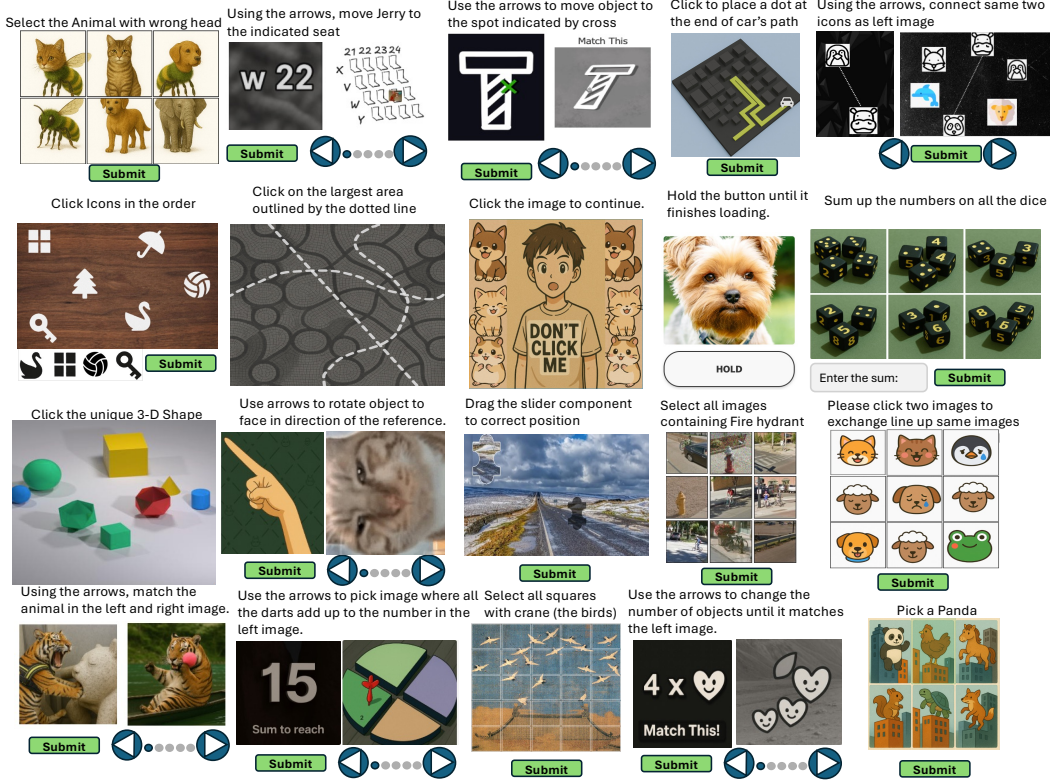

Figure 9: **More Examples of Open CaptchaWorld.**

B   MLLM Models Performance Analysis on Different CAPTCHA Types

Table 2 presents a capability support matrix that summarizes whether each multimodal agent successfully solved at least one instance of each CAPTCHA type in our benchmark. A "✓" indicates that the model demonstrated at least partial success on that type, while "✗" indicates complete failure across all test instances. This table helps visualize the distribution of strengths and weaknesses among different MLLM agents. We observe that certain tasks, such as *Image Recognition*, *Image Matching*, and *Select Animal* are universally solved by nearly all models, suggesting they rely primarily on basic visual grounding or object recognition. In contrast, tasks requiring spatial manipulation (*Slide Puzzle*), counting (*Dice Count*), dynamic control (*Hold Button*), or path reasoning (*Path Finder*) remain unsolved by all models.

Notably, Openai-o3 shows the broadest support across CAPTCHA types, including moderate success on tasks like *Patch Select*, *Dart Count*, and *Rotation Match*, which require multi-step reasoning or spatial judgment. Meanwhile, other models like Claude-3.7-Sonnet show isolated strengths, for instance, uniquely solving *Hold Button* and *Bingo*-type tasks, indicating variation in architectural strengths or alignment training. This breakdown reinforces that existing MLLM agents exhibit significant variance in cross-task generalization and often struggle with interaction-heavy or arithmetic-based challenges. The table serves as a diagnostic tool for future model benchmarking and agent specialization analysis.

Table 2: Support of different models on various types of CAPTCHA tasks.

	Openai-o3	Openai-o1	GPT-4.1	GPT-4o	Gemini2.5-Pro	Claude-3.7-Sonnet	Claude-3.5-Haiku	Claude-3.5-Sonnet	DeepSeek-V3
Dice_Count	✗	✗	✗	✗	✗	✗	✗	✗	✗
Geometry_Click	✗	✗	✗	✗	✗	✗	✗	✗	✗
Rotation_Match	✓	✗	✗	✗	✗	✗	✗	✗	✗
Slide_Puzzle	✗	✗	✗	✗	✗	✗	✗	✗	✗
Unusual_Detection	✗	✗	✗	✗	✗	✗	✓	✗	✗
Image_Recognition	✓	✓	✓	✓	✓	✓	✗	✗	✗
Bingo	✗	✗	✗	✗	✗	✓	✓	✗	✗
Image_Matching	✓	✓	✓	✓	✓	✗	✓	✗	✗
Patch_Select	✓	✓	✗	✗	✓	✗	✗	✗	✗
Dart_Count	✓	✗	✗	✗	✗	✗	✗	✗	✗
Object_Match	✓	✓	✓	✗	✓	✗	✗	✓	✓
Select_Animal	✓	✓	✓	✓	✓	✓	✓	✓	✓
Coordinates	✗	✗	✗	✗	✗	✗	✗	✗	✗
Path_Finder	✗	✗	✓	✓	✗	✗	✗	✗	✗
Place_Dot	✗	✗	✗	✗	✗	✗	✗	✗	✗
Connect_icon	✗	✗	✗	✗	✗	✗	✗	✗	✗
Click_Order	✗	✗	✗	✗	✗	✗	✗	✗	✗
Hold_Button	✗	✗	✗	✗	✗	✓	✗	✗	✗
Misleading_Click	✗	✗	✗	✗	✗	✗	✗	✗	✗
Pick_Area	✗	✗	✗	✗	✗	✗	✗	✗	✗

C Reasoning Depth Annotation Guidelines

To estimate the **Reasoning Depth** of a CAPTCHA puzzle, we define a checklist of atomic reasoning and interaction steps that a human must perform. Each step corresponds to a discrete visual, cognitive, motor, or state-transition operation. A CAPTCHA's total reasoning depth is computed by counting how many of these atomic steps are required to solve it correctly. Each satisfied atomic step contributes a depth of +1.

Annotators are instructed to use the following table as a reference. For every puzzle analyzed, they should determine which of the atomic steps are involved, and report the total reasoning depth accordingly. For transparency, all annotations must be accompanied by justifications that cite specific steps from the table.

> **Prompt to Estimate CAPTCHA Reasoning Depth**
>
> To estimate the reasoning depth of a CAPTCHA puzzle, use the following rules as checklist: {Rules}. The goal is to assess how many discrete reasoning or interaction steps a human would need to solve the puzzle. Use the provided checklist as a reference, but do not rigidly count checklist items. Instead, reflect on the actual sequence of cognitive and motor steps needed to reach the solution. For each estimated depth, clearly explain your reasoning.

Figure 10: Prompt for estimating CAPTCHA Reasoning Depth.

> **Prompt to Test Broswer Use Agents on Open CaptchaWorld**
>
> You are an autonomous CAPTCHA-solver for the **Open CaptchaWorld** webpage. Go to http://localhost:5001/ and solve the CAPTCHA challenges as many as you can. Notice, You may need to click some buttons to solve the captcha.

Figure 11: **Prompt to Browser Use Agents for testing on Open CaptchaWorld.**

Table 3: Checklist of Atomic Steps for Reasoning Depth Estimation

Category	Atomic Step Description
Visual (V)	Locate a single target object class
	Read an entire multi-character CAPTCHA string
	Detect orientation of one jigsaw tab
	Identify a color-coded region
	Recognize a specific symbol or emoji
	Count objects in a scene
	Spot the difference between two panels
	Read numeric code displayed on a dial
	Interpret a legend or map key
	Detect newly revealed hint after a state change
Cognitive (C)	Decide a subset satisfying a logical rule
	Plan the order of assembling pieces
	Infer a hidden sorting principle
	Translate a textual instruction into an action set
	Choose the optimal path in a maze
	Determine the required rotation angle before acting
	Resolve ambiguity between visually similar targets
	Memorize a short cue for later recall
	Select the correct tool among many options
	Apply elimination logic to narrow down choices
Motor (M)	Single left-click on a target
	Bulk-select multiple tiles after a single decision
	Drag-and-drop one piece (grab $\rightarrow$ release)
	Continuous slider movement to a target position
	Rotate a dial or knob in one continuous motion
	Type a full code in one uninterrupted entry
	Draw a single straight line
	Resize a bounding box once
	Check or uncheck a checkbox
	Press-and-hold a button until success
State Reveal (V)	Observe the puzzle state after an automatic change