OSWORLD-PRO: PROCESS-BASED EVALUATION FOR COMPUTER USE AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluation of Computer-Use Agents (CUAs) is often limited to the final deliverables they create (at the end of hundreds of steps) and assessed with functional verifiers, as seen in OSWorld. However, such evaluation of end-state performance lacks transparency into *how and why* agents fail in various tasks, obfuscating critical insight for subsequent improvement. For instance, agents that err during keyboard inputs would require a different mitigation strategy from those that fail to precisely provide click-based inputs on the graphical UI. We introduce OSWorld-Pro: a set of over 300 tasks containing over 2800 subgoals to enable the procedural evaluation of CUAs grounded in over 67,000 human annotations. We use robust human-aligned LLM-Judges to evaluate the fulfillment of OSWorld-Pro subgoals and thereby reveal the progress that models make throughout a series of sequentially dependent subgoals. Our findings reveal that OSWorld-Pro is challenging even for state-of-the-art LLMs, with top performers like Claude Opus 5 achieving only 75.7% vs. 83.4% on OSWorld. Furthermore, we identify critical process-focused failure modes of various models (e.g. subgoal-irrelevant actions and click-based mistakes) to provide insights to improve performance and efficiency of CUAs.

1 INTRODUCTION

Humans use computers to tackle long-horizon tasks involving many interdependent subgoals, tracking their progress through intermediate milestones rather than relying solely on final outcomes. For example, in a machine learning research project such as the one presented in this paper, researchers assess progress by monitoring data collection, iterating on computational experiments, and visualizing experimental results. In such situations, humans often do not rely on final outcomes alone (such as completed paper manuscripts) and instead utilize the status of various constituent subgoals to determine how these projects are progressing. Using the same vein of thought, we believe that

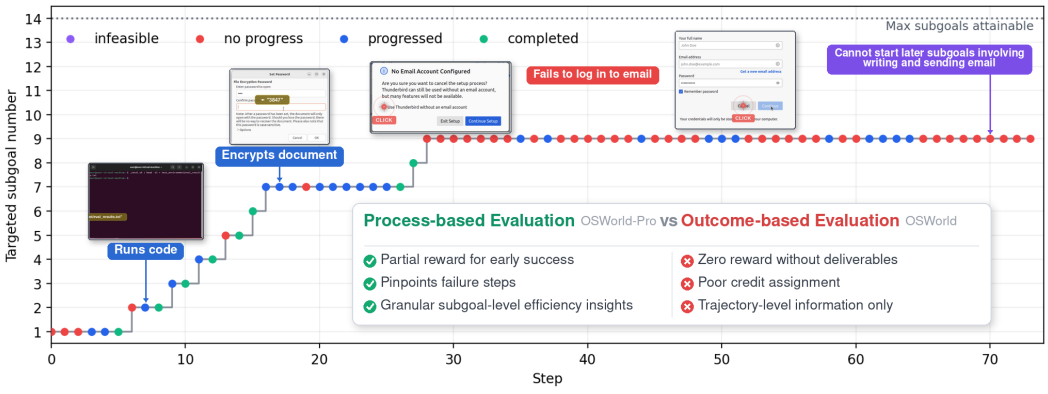


Figure 1: OSWorld-Pro provides partial reward for early success without requiring final deliverables, pinpoints specific failures and offers granular information on progress through sequentially dependent subgoals. Such advantages of Process-based evaluation for Computer Use Agents (CUAs) complements the limitations of outcome-based evaluations such as OSWorld.

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.

OSWorld-Pro Example (vs. OSWorld Example at bottom)

Category: Coordination (between $\geq$ 4 different applications)
Goal: Open Visual Studio Code from the sidebar. Create a new Python file named `calculate.py` in the Home directory that contains a function to calculate the factorial of a number and print the result. Run the script in the terminal with an input of 5. Then copy the output from the terminal and paste it into LibreOffice Writer. Format the pasted text as bold with font size 14, and save the document as `factorial_result.docx` on the Desktop.

Subgoals: *[App used]*

1. Open Visual Studio Code from the sidebar. *[OS]*
2. Create `calculate.py` in the Home directory. *[VS Code]*
3. Write a function that calculates the factorial of a number and prints the result. *[VS Code]*
4. Run `calculate.py` in the terminal with an input of 5. *[Terminal]*
5. Copy the output from the terminal. *[Terminal]*
6. Paste the copied output into LibreOffice Writer. *[LibreOffice Writer]*
7. Format the pasted text as bold with font size 14. *[LibreOffice Writer]*
8. Save the document as `factorial_result.docx` on the Desktop. *[LibreOffice Writer]*

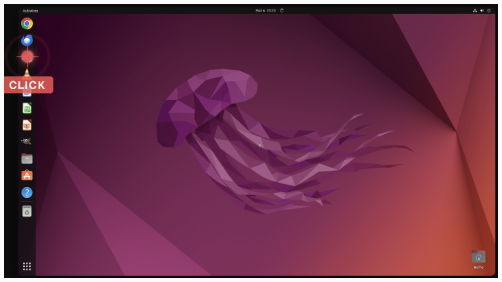
Steps: (e.g. Step 1)

Action:
`pyautogui.click(0.019, 0.184)`

Reasoning:
... The first step is to click on this icon to open Visual Studio Code. Let me click on the VS Code: icon to start the process.

Human Annotation:

Subgoal(s) targeted: **{1}**
Subgoal(s) feasibility: ✓
Subgoal(s) progression: ✓
Subgoal(s) completion: ✓



OSWorld Example: Make the line spacing of first two paragraph into double line spacing *[LibreOffice Writer]*

Figure 2: OSWorld-Pro Example compared with OSWorld Example. More examples in § A.

process-based evaluations of Computer-Use Agents can complement existing benchmarks based on outcome-based evaluations (e.g. OSWorld).

Outcome-based evaluations were utilized with great success on math capabilities in terms of GSM8K (Cobbe et al., 2021) and AIME 25 (White et al., 2025) and later in coding environments (using unit tests) such as LiveCodeBench (Jain et al., 2024) as well as scientific question answering in GPQA (Rein et al., 2023), MMLU-Pro (Wang et al., 2024), and HLE (Phan et al., 2025). Outcome-based evaluation was also applied on CUA capabilities in works such as the widely-adopted OSWorld (Xie et al., 2024) and OSUniverse (Davydova et al., 2025). These works construct functional verifiers against the final deliverables of the tasks and measure the success of agents based on whether the deliverables match various aspects of reference files. Beyond desktop GUI agents, there have also been adjacent work relating to Android GUI (Rawles et al., 2025) as well as tool calling ability with model context protocol (Jia et al., 2025).

However, such outcome-based evaluation approaches for CUAs have some limitations. First, they are unable to discriminate between trajectories with different progression in tasks that have yet to create final deliverables. For instance, in Fig. 1, final deliverables will not be available whether the agent fails at the first subgoal or the ninth subgoal and hence the agent will receive a zero for both trajectories even though the agent has progressed much further in the later trajectory. Such poor discernibility becomes more critical in long-horizon tasks (e.g. requiring hundreds of steps and therefore have a high likelihood of failure prior to final deliverable creation). Second, as CUAs become stronger, they will also be more capable in terms of reward hacking, which means that these agents can get to the correct final outcome through undesirable ways. The recent OpenAI security incident (OpenAI, 2026b) highlights how strong agents can literally hack third-party servers to obtain restricted information (i.e. answer keys) in order to do well on outcome-based evaluations. Finally, outcome-based evaluations do not provided fine-grained information on the contribution of individual

steps within the agent trajectory, which can be useful to assess how efficiently agents advance on long-horizon tasks. Process-based evaluation addresses these limitations by focusing not only on *what* final outcomes CUAs arrive at but also *how* they get there.

To design process-based evaluation for CUAs, we draw inspiration from PRM-800k (Lightman et al., 2023) and ProcessBench (Zheng et al., 2025), which are process-based evaluations for math capabilities. Specifically, these benchmarks break down reasoning on solving math problems into distinct steps. Then, they seek to identify where errors first occur in the reasoning chain. CUA tasks tend to be much more open-ended compared to math tasks in PRM-800K and ProcessBench, as CUA tasks often have multiple approaches to reach the required goal. Therefore, we adapt ideas from works on rubrics (Gunjal et al., 2025; Arora et al., 2025; Wang et al., 2026). Specifically, we break down the overarching task goal into subgoals that can be individually assessed for completion. However, unlike rubrics, which are typically requirements that models can fulfill independently, subgoals in many CUA tasks are dependent on each other. This means that an earlier subgoal has to be completed before moving to the next subgoal.

To support process-based evaluation for CUAs, we present OSWorld-Pro: a benchmark with over 67, 000 step-level human annotations across more than 2800 progressive subgoals in over 300 long-horizon CUA tasks. The main features for OSWorld-Pro are:

1. **Long-Horizon**: Tasks have an average of 9.2 sequentially dependent subgoals per task, for which earlier subgoals have to be completed prior to attempt subsequent ones. This is in contrary to benchmarks like OSWorld where tasks mostly have a single subgoal (see example in Fig 2) and ChainWorld (Siu et al., 2026), which chains up multiple loosely-connected OSWorld tasks and do not reflect the interdependent nature of subgoals in real-world long-horizon tasks.

2. **Challenging**: Tasks require an average of 3.45 unique apps to complete, which is substantially more compared to OSWorld at 1.34. In addition, we include rare apps such as Videos, Archive Manager and LibreOffice Draw and uncommon Linux and GUI distributions - beyond Ubuntu with GNOME - (e.g. AlmaLinux and MATE) not found in OSWorld to test generalization capabilities. Among top performing models, Claude Opus 5 only reaches 75.7% vs. 83.4% on OSWorld (XLANG-Lab, 2025).

3. **Fine-grained**: Beyond providing an aggregate metric that shows how well an agent performs, OSWorld-Pro allows users to understand the type of actions that it commonly fails on, how efficient it is in completing subgoals over its trajectories and how it react when facing infeasible subgoals, as common in real-world tasks. For instance, Claude Opus 5 was shown to occasionally engage in subgoal-irrelevant actions (sometimes for over 50-steps) despite completing all subgoals.

2 OSWorld-Pro Overview

OSWorld-Pro contains 67,264 human-annotated labels across 305 distinct tasks with 2814 progressive subgoals. These tasks span three categories: Diversity (117 tasks), covering less commonly benchmarked applications; Coordination (109 tasks), requiring coordination across ≥ 4 applications; and Robustness (79 tasks), testing generalization across Linux distributions and graphical interfaces. Specifically, samples contains agent trajectories on various tasks as well as step-level human annotations on how each step targets various subgoals, determine their feasibility and monitor their progression and final completion. While the sheer number of tasks (305) is comparable to the 369 tasks in OSWorld (Xie et al., 2024) and 160 tasks in OSUniverse (Davydova et al., 2025), each OSWorld-Pro task has 2 orders of magnitude more granular human-annotated verification signals (collected over > 5000 person-hours) compared to alternatives. We show an example in Fig. 2, a visualization of data distribution in Fig. 3, and further descriptive statistics in §B.

3 Data Collection

Annotator Recruitment To ensure annotation quality, we select qualified annotators through screening, training and pairing each annotator with an experienced reviewer. Annotators with Bachelors' degrees or higher, as well as ≥ 6 months of experience working on Computer-Use Agent annotation are recruited and managed by our vendor. Prior to their inclusion into the project, we check that they pass tests on English capabilities, Linux proficiency (post mandatory training) and understanding of the annotation workflow (through two sample tasks reviewed against golden references). Each annotator works through the entire agent trajectory of a task and is supported by a

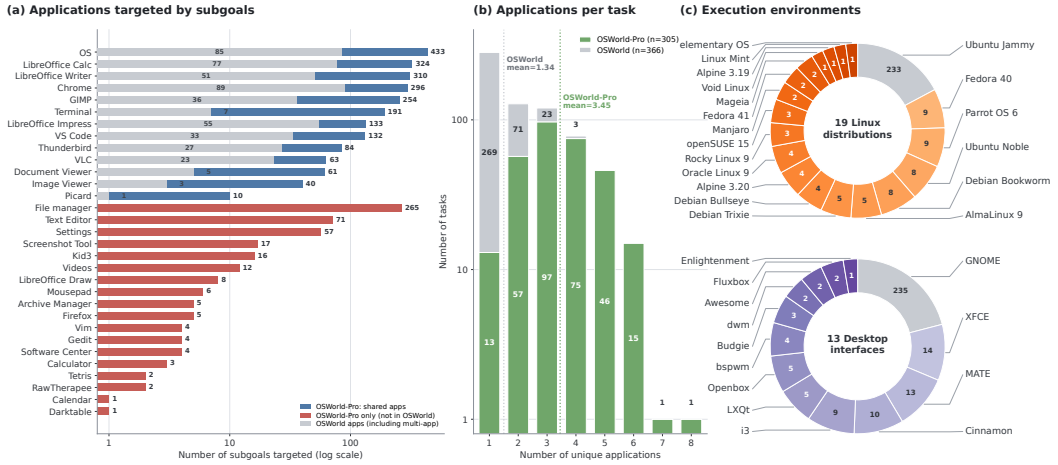


Figure 3: OSWorld-Pro Data Distribution: Compared with OSWorld, OSWorld-Pro covers a wider diversity of applications (31 vs 13), requires coordination between more applications in a single task (mean = 3.45 vs 1.34) and can show robustness of agents in more environments (19 Linux distributions + 13 Graphical interface vs Ubuntu Jammy with GNOME only) .

reviewer (with substantial annotation experience) to iteratively review the annotator's work and to provide feedback for improvement where helpful. Across 305 tasks, 25 annotators and reviewers from 5 countries were involved. Further details on annotator recruitment in §C.

Task Curation We generate tasks using the approach described in ProCUA-SFT (Jung et al., 2026) for human annotation. Specifically, for the Diversity category, we identify tasks containing applications that are underrepresented in existing computer-use benchmarks (e.g. Videos, Archive Manager and LibreOffice Draw). For the Coordination category, we select tasks that require coordinated use of at least four applications. By comparison, tasks in OSWorld involve at most four applications. For the Robustness category, we generate additional tasks using the ProCUA-SFT approach[1] in environments that differ from the default Ubuntu/GNOME setup. These environments include Linux distributions such as Fedora, Alpine, and AlmaLinux, and graphical interfaces such as bspwm, Xfce, and LXQt, as detailed in Fig. 3. These tasks evaluate how well models generalize across Linux distributions and graphical interfaces.

Subgoal Decomposition ProCUA-SFT tasks only contain an overarching task goal as well as the set of apps required for the task goal. We break down the tasks into atomic subgoals that can be independently assessed as well as a specific application required for each subgoal. Specifically, we prompt DeepSeek-V4-Pro (DeepSeek-AI et al., 2026) using the prompt template in §E.

Human Annotation Our annotation workflow consists of task validation, annotator assignment, step-level labeling, and lastly independent and interactive review. **(1) Task validation** Prior to human annotations, our vendor inspects and removes tasks that have unclear or under-specified goals or have safety concerns. In addition, the subgoals and application required by each subgoal are also manually inspected and corrected where appropriate. In addition, we skip tasks where subgoals are not sequentially dependent on one another (i.e. only include tasks where earlier subgoals need to be completed before later subgoals). In doing so, we avoid overly simple tasks with multiple unrelated subgoals (e.g. open a video file, then open an unrelated text document) to focus on long-horizon tasks that are more challenging and realistic. **(2) Annotator assignment** We assign tasks to technical or non-technical annotator pools based on the applications and skills involved. Tasks requiring coding knowledge, such as those involving coding in VS Code or the terminal, are assigned to technical annotators. **(3) Step-level labeling** Annotators label each step of pre-generated model trajectories to identify the subgoal(s)[2] that a step targets. In some environments, we notice that the targeted subgoal is not feasible. For example, a subgoal may require selecting an option from the file menu that does

[1]We use Kimi-K2.6, available in May 2026, instead of Kimi-K2.5.

[2]This is almost always a single subgoal but $<2\%$ of steps do effectively target two or more subgoals.

not exist. Therefore, we also ask annotators to indicate whether the targeted subgoal(s) are feasible. Following OSWorld (Xie et al., 2024), we keep tasks with infeasible subgoals in order to understand what models would do in similar real-world tasks. If it is feasible, annotators also indicate if the current step makes progress on and completes the subgoal. **(4) Independent and interactive review** Inspired by the annotation workflow in ProfBench (Wang et al., 2026), our workflow involves having a reviewer to iteratively provide feedback for the annotator to modify their annotations. However, because the task is time-consuming (annotators spend between 5 to 20 hours per task depending on the number of steps taken), we found it challenging for reviewers to provide comprehensive feedback, despite their strength in precisely pointing out inadequacies. Therefore, we modify our workflow based on the approach taken in HelpSteer3 (Wang et al., 2025b) such that the reviewer has to first independently annotate the task before providing feedback to the annotator with the annotation platform (Super Annotate) highlighting the differences in initial annotations. Annotators were not allowed to use LLMs during the annotation task, with extensive checks to ensure compliance (e.g. based on repetition patterns and annotation durations). Full annotation guidelines are in §D.

4 Can LLMs effectively judge like humans do?

Task Formulation Given the high cost of having humans evaluate agent trajectories, many works (Starace et al., 2025; Arora et al., 2025; Wang et al., 2026) have turned to using LLM-Judges to proxy human judgments. Computer Use Agent trajectories are particularly challenging to evaluate because they are path-dependent across long-horizons (e.g. across hundreds of steps) while evaluation of agent outputs (e.g. in OSWorld) is only state-dependent, without considering how the agent arrived at the final state. More precisely, the role of the LLM Judge is to identify which subgoal(s) are targeted by every step of the agent trajectory, whether that subgoal is feasible at the step and if so, whether the agent makes progress and/or completes the subgoal at that step.

4.1 Evaluation

Agreement with Human Annotations To evaluate LLM-Judges, we use Macro-F1 based on the human-labeled ground-truth and the model-predicted label as used by ProfBench (Wang et al., 2026) and PaperBench (Starace et al., 2025). Subgoal(s) targeted is first binarized into whether each subgoal is targeted by a particular step while the other fields (feasibility, progression and completion) are innately binary. Given the dependent nature of the fields, we consider values in fields iff the pre-requisite field was correct. This means we consider feasibility iff the correct set of subgoals were identified, progression iff feasibility was correct and completion iff progression was correct.

Inference Setup We ran early experiments with GPT-5.6-Luna (with default medium reasoning), a lightweight, affordable and feasible at high concurrency. The complexity of judgment task (with hundreds of step over tens of subgoals) required extensive explorations in LLM-judge design, which we discuss in §F. Our final design involves evaluating an entire trajectory in a single API request. Because this requires sending a payload of up to hundreds of screenshots in a single request (requiring 500 MBs), we only found success in using OpenAI GPT-5.6 models while others (e.g. Claude, Gemini and Open Models) raised various errors relating to payload size, image quantity or context window. Cost estimation method is detailed in §G.

Performance at Different Levels In addition to Macro-F1 at the step level, we want to understand how well the LLM-Judge can predict human annotations at three complementary levels. First at the step level, we calculate a simple mean across macro-F1 across subgoal {targeted, feasibility, progression, completion}. Next at the subgoal level, we tabulate the proportion of subgoals that have their final completion status across all steps in a trajectory predicted correctly. Subgoals are considered completed iff at least one step has the targeted subgoal labeled as completed. Finally, at the task level, we calculate an overage task performance based on the percentage of subgoals completed. We calculate a mean absolute error (MAE) between the human-annotated and model-predicted performance and take the average across all tasks. Finally, we calculate 1-MAE to make the metric higher as better and improve readability.

Reliability of Human Annotations To understand how reliable the Annotators' labels are, we compare them to the Reviewer's initial annotations prior to see annotations from the Annotator

and providing feedback to the Annotator. This provide a measure of how much two independent annotators would agree. We find excellent agreement (Cohen's $\kappa > 0.8$) across all aspects with 0.987 for subgoal targeted, 0.847 for feasibility, 0.869 for progression and 0.940 for completion.

4.2 RESULTS

Table 1: Evaluation of LLM-Judges. Higher is better for Macro-F1 and Performance at different levels while lower is better for Tokens.

	Step Agreement w. Human Labels (Macro-F1) ↑				**Performance at Different Levels ↑**			**Tokens ↓**		
Model	Target	Feasible	Progress	Complete	Step (Mean)	Subgoal	Task (1-MAE)	In/Task	Out/Task	Total $
Human Performance	98.4	91.6	94.1	97.6	95.4	95.6	96.0	-	-	-
LLM Judge										
OpenAI/GPT-5.6-Sol										
- max	97.0	61.9	74.7	94.3	82.0	94.1	93.0	144566	12004	249.60
- xhigh	97.2	65.3	71.6	93.8	82.0	93.9	93.0	144566	7072	219.51
- high	97.0	65.8	69.8	93.3	81.5	93.7	92.6	144566	4900	206.26
- medium	96.8	64.1	68.1	91.3	80.1	92.9	92.5	144566	3631	198.52
- low	95.8	65.9	65.2	86.8	78.4	89.1	90.5	144566	2706	192.88
- none	94.5	67.5	69.5	67.8	74.8	83.8	86.5	144566	2069	188.99
OpenAI/GPT-5.6-Terra										
- max	97.6	62.0	69.2	94.8	80.9	93.9	92.4	144566	18266	155.04
- xhigh	97.2	64.5	63.9	91.6	79.3	92.6	90.9	144566	7192	114.51
- high	95.9	63.8	62.3	86.2	77.1	92.2	91.3	144566	4168	103.44
- medium	94.9	67.4	59.6	82.9	76.2	89.6	90.7	144566	2925	98.89
- low	94.9	66.0	61.3	80.0	75.6	89.1	89.8	144566	2609	97.74
- none	93.3	65.2	67.6	82.8	77.2	91.5	91.3	144566	1961	95.36
OpenAI/GPT-5.6-Luna										
- max	97.7	58.9	52.8	93.3	75.7	91.2	89.6	144566	18111	15.45
- xhigh	97.1	60.4	51.0	89.6	74.5	87.3	87.4	144566	11736	13.11
- high	96.5	58.8	49.8	86.5	72.9	84.0	84.9	144566	6713	11.28
- medium	95.6	57.3	47.6	84.6	71.3	85.1	87.5	144566	2942	9.90
- low	94.9	54.0	46.4	87.0	70.6	88.5	89.0	144566	2309	9.66
- none	90.4	58.9	53.6	81.2	71.0	84.3	86.7	144566	1932	9.53

Which aspects are LLM Judges lacking compared to humans? The top performing model GPT-5.6-Sol with max reasoning effort approaches Human Performance on identifying targeted subgoals (97.0 vs 98.4%) and comes close on predicting subgoal completion at the step, subgoal and task levels (within 3.3% absolute different from humans) as shown in Tab. 1. However, it substantially lags behind on predicting subgoal feasibility (61.9 vs. 91.6%) and progress (74.7 vs. 94.1%), which also have lower agreement rates between independent human annotators (Cohen's κ=0.847-0.869 vs 0.940-0.987). We suspect it is because these fields are more subjective as it might not always be clear whether a task is infeasible or a practical method to achieve a goal has not yet been found. For example, when locating an option under a menu with various unexpanded sub-menus, a feasible subgoal might appear infeasible until the menu option is discovered (or vice-versa). Similarly, whether a step meaningfully helps to progress toward a subgoal might at times be ambiguous. For instance, if a step modifies a piece of code to add a new functionality but results in a bug, a judge can reasonably justify both progress=No or progress=Yes depending on what the focus is on.

Does Model Size Matter? Within the GPT-5.6 model family, larger models generally do better than smaller models but the improvement plateaus. For instance, the step level performance (Mean of 4 Macro-F1s) in Tab. 1 substantially increase from 75.7 (Luna) to 80.9 (Terra) and then only slightly increases to 82.0 (Sol) while at Max reasoning. This rate of increase is roughly inline with gains in total cost, which first increases 10x and then rises only 1.6x. When constrained on API cost, increasing the reasoning effort on a smaller model often beats a lower reasoning effort on a larger model at a lower cost. For instance, GPT-5.6-Terra Max slightly outperforms GPT-5.6-Sol Medium at 22% lower price while GPT-5.6-Luna Max generally matches GPT-5.6-Terra Low at one-sixth cost.

How much should a model think? Models generally perform better at higher reasoning efforts across most metrics. However, there are two major exceptions to the rule. Models sometimes perform better with no reasoning compared to low reasoning. This might be because given models insufficient thinking tokens might artificially cut short its explicit thinking process, while non-reasoning model are unaffected in their implicit, latent reasoning through hidden layers (Li et al., 2025). Subgoal Feasibility performance generally also lowers with higher reasoning effort (on Terra and Sol). Such a correlation is also observed in terms of the likelihood for models to predict feasibility=No. The proportion of feasibility=No decreases from 9.7% at GPT-5.6-Sol None to 5.1% at GPT-5.6-Sol Max while human-annotated ground-truth is at 15.3%. This suggests that increased reasoning effort results in models over-estimating the feasibility of tasks and hence resulting in lower macro-F1.

5 BENCHMARKING MODELS ON OSWORLD-PRO

Table 2: OSWorld-Pro Performance across select Closed-Source and Open-Weight models.

Model	% of Tasks with All Subgoals Completed ↑				Task-Level Efficiency ↓		
	Overall	Diversity	Coordination	Robustness	Steps	Output Tokens	Token $
Closed-Source							
Claude Opus 5 Max	75.7	81.2	70.6	74.7	76.0	23492.6	3.36
Claude Opus 4.8 Max	77.7	78.6	83.5	68.4	67.1	67069.6	5.60
Claude Opus 4.7 Max	76.7	78.6	80.7	68.4	98.8	30886.2	4.95
Claude Sonnet 5 Max	76.4	82.1	81.7	60.8	108.7	40804.6	2.47
GPT-5.6-Sol Max	74.8	76.1	70.6	78.5	53.3	39595.2	9.56
GPT-5.6-Terra Max	70.8	74.4	71.6	64.6	54.9	43777.5	4.98
GPT-5.6-Luna Max	74.4	74.4	74.3	74.7	60.2	41377.1	0.51
Gemini-3.8-Flash High	59.0	65.8	59.6	48.1	71.7	18966.8	2.81
Open-Weight							
Kimi K3 Max (2.8T)	39.3	46.2	43.1	24.1	42.2	125870.7	5.51
Minimax M3 Xhigh (428B)	28.9	40.2	30.3	10.1	201.3	36952.1	1.04
Qwen 3.8 Flash Next Xhigh (125B)	55.1	66.7	58.7	32.9	75.0	26940.4	0.20
Qwen 3.5 122B Xhigh	16.7	32.5	11.0	1.3	127.2	42316.4	0.68
Qwen 3.8 27B Xhigh	32.1	43.6	33.9	12.7	48.5	15177.7	0.15
Qwen 3.6 27B Xhigh	10.2	20.5	6.4	0.0	31.5	7075.7	0.16

Task Formulation Following Xie et al. (2024) and Jung et al. (2026), we formalize the task as given a goal alongside an computer-use Linux environment with Graphical UI (that it can "see" via screenshots), the agent (VLM with harness defined in OSWorld) should generate a trajectory that addresses the goal. Subsequently, we use the best-performing GPT-5.6-Sol Max judge from §4 to evaluate whether steps fulfill various subgoals. Finally, we report the percentage of tasks across each category that complete all subgoals deemed feasible by human annotators. We believe that GPT-5.6-Sol Max judge is an adequate proxy of human judgments as it matches human judgment in 94.1%, closely trailing independent reviewers at 95.6% from Tab. 1. In addition, GPT-5.6-Sol Max scores itself lower than 4 other models, alleviating our initial concerns over potential self-preference bias. Our evaluation setup largely follows OSWorld GitHub (XLANG-Lab, 2024) and evaluates only vision-language models with publicly available harnesses there (detailed in §G).

OSWorld-Pro is Challenging Overall, Claude Opus 4.8 with Max effort achieves the best performance in Tab. 2 at 77.7% Overall, which means that OSWorld-Pro is substantially more challenging than OSWorld where the top Opus model scores 83.4% (XLANG-Lab, 2025). OSWorld-Pro is particularly challenging for open-weight models with the top model only achieving 55.1% whereas open-weight models scores >80% on OSWorld. For a matched model, Minimax M3 scores only 28.9% on OSWorld-Pro but 75.2% on OSWorld. This indicates that OSWorld-Pro can be a good target for open-weight models to hill-climb against, without being saturated or overly-difficult such that improvements are hard to achieve.

Among open-weight models, scores are generally highest on the Diversity category (least challenging), followed by Coordination and finally Robustness (most challenging). This means that open-weight models generalize best to rare applications, moderately to longer-horizon workflows requiring coordination between ≥ 4 apps and worst on different Linux distributions and graphical interfaces. One possible explanation for the poor performance relating to generalizations across different Linux and graphical UI environment is the general lack of such data within training environments, due to the limited commercial advantage of improving on them (given how esoteric they are in real-world work settings). Therefore, models are forced to generalize out-of-distribution from Ubuntu/GNOME environments that are more common in training data (Wang et al., 2025a; Jung et al., 2026).

Does Parameter-Scaling Work? To understand the role that model-size plays on OSWorld-Pro performance, we conduct some analysis across both closed-source and open-weight models. Closed-source models in general perform better compared to open-weights one. As the parameter counts for closed models are not publicly reported, we use the observation that within the same model family, more expensive models by API pricing are likely to be larger (e.g. GPT-5.6 Sol > Terra > Luna, Opus 5 > Sonnet 5). We find no obvious evidence that larger parameter count alone translates to better performance on OSWorld-Pro given that Sonnet 5 does better than Opus 5 (but worse than Opus 4.7 and 4.8) and GPT-5.6 Luna does better than Terra but worse than Sol.

However, we observe that smaller model typically have a larger step count (e.g. Luna with 60.2 steps vs. Sol at 54.9 steps; Sonnet 5 at 108.7 steps vs. Opus 5 at 76.0 steps). This suggests that smaller models can use a greater number of steps to partially compensate for model capacity proxied via parameter count. On open-weight models, parameter count could potentially contribute some improvement as Qwen 3.8 Flash Next (125B) does better than Qwen 3.8B 27B at 55.1% vs. 32.1% from the same model family but it could be confounded by the difference in model architecture (Dense vs MoE). Furthermore, models with similar parameter count (e.g. Qwen 3.8 27B vs. Qwen 3.6 27B) have drastic different performance at 32.1% vs. 10.2% as with Qwen 3.8 Flash Next 125B and Qwen 3.5 122B (55.1% vs. 16.7%), suggesting that training recipes could influence performance more compared to model size alone.

Task-Level Efficiency is critical in real-world tasks (beyond task completion) as it influences user experiences in terms of latency and additional cost. There are three complementary perspectives for users who care about different aspects: steps, output tokens and token cost. Latency-sensitive users should focus on a combination on the number of steps as well as the output-tokens while token costs should be the priority for cost-sensitive users. One observation is that some models have low step count but high output tokens (e.g. Kimi K3 with 125 thousand tokens in only 42.2 steps). A possible explanation is that the pyautogui library that evaluation depends on supports multiple actions per step and hence models like Kimi K3 does fewer steps overall but seeks to perform more in each step, which requires more response tokens (of which, many are used for thinking). Another observation is that cost per task can differ by 20x across models with similar performance ($0.51 for GPT-5.6 Luna vs. $9.56 for GPT-5.6-Sol), suggesting that good performance can be balanced with affordability.

Table 3: Analysis of Subgoal Progression Likelihood by Action Category as well as Efficiency

Model	**Subgoal Progression Likelihood ↑**							**% Subgoals ↑**	**Subgoal Efficiency ↓**		
	All	Click	Drag +Move	Keyboard	Scroll	Execution Control	Others	Completed (Feasible Only)	Successful Effort	Failure Persistence	Infeasible Persistence
Claude Opus 5 Max	86.5	89.0	73.9	92.9	92.9	78.3	83.9	85.6	7.2	10.5	9.5
Claude Opus 4.8 Max	89.6	92.5	74.3	92.6	93.8	84.5	85.4	86.7	6.7	12.1	12.1
Claude Opus 4.7 Max	80.7	86.3	41.0	89.2	84.1	69.9	73.9	91.0	8.7	18.9	32.2
Claude Sonnet 5 Max	85.5	88.0	44.9	93.0	91.1	73.4	81.8	89.2	9.9	28.7	39.8
GPT-5.6-Sol Max	78.9	86.9	66.5	73.7	74.6	-	71.7	93.1	4.9	8.2	17.8
GPT-5.6-Terra Max	79.3	86.9	61.2	74.4	63.1	-	73.8	92.8	5.3	9.1	12.1
GPT-5.6-Luna Max	77.7	84.7	57.3	73.1	74.2	-	79.2	93.2	5.6	10.7	26.8
Gemini 3.8 Flash High	74.4	83.1	76.8	73.4	-	82.1	64.4	67.9	8.1	18.6	20.5
Open-Weight											
Kimi K3 Max (2.8T)	54.5	41.9	38.7	54.8	48.1	65.3	58.4	58.0	5.1	12.5	20.2
Minimax M3 Xhigh (428B)	49.7	39.0	24.2	78.2	66.6	40.4	57.2	52.5	21.2	64.3	80.7
Qwen 3.8 Flash Next Xhigh (125B)	73.4	72.4	53.9	76.6	78.6	70.7	50.2	78.2	7.3	19.9	22.5
Qwen 3.5 122B Xhigh	32.3	29.6	11.3	41.4	34.7	35.4	5.9	42.7	6.4	62.8	24.3
Qwen 3.8 27B Xhigh	80.0	82.7	51.6	79.3	70.2	84.8	52.2	54.9	6.3	12.3	32.7
Qwen 3.6 27B Xhigh	65.4	65.9	55.6	63.1	58.8	73.7	58.3	29.0	5.5	9.9	12.5

6 Analysis: What kind of actions and subgoals trip up models?

Process-based evaluations (like OSWorld-Pro) have unique advantages in revealing insights on efficiency and failures over an agent trajectory that outcome-based evaluation like OSWorld cannot.

Action-Level Failure Analysis To identify which action types are most prone to failure for each model, we measure action-level progression likelihoods across agent trajectories. Specifically, we calculate the proportion of steps that make progress toward the subgoals. A step is considered if it targets at least one subgoal and the targeted subgoals are feasible. We exclude action categories that were observed fewer than 5 times to reduce noise from limited observations. Higher scores are better. Most models perform well on click operations, with the exception of Minimax M3 and Kimi K3. A closer look at Minimax M3 trajectories show frequent failures in basic operations such as clicking on the correct coordinates to perform an action (e.g. wanting to close a window but not clicking on the x button as shown in Fig. 4) while stronger models such as Claude and GPT-5.6 rarely commit such errors. This is reflected in the difference in progression likelihood for click operations on Tab. 3 for these models (39.0-41.9%) vs. stronger models (72.4-92.5%). In addition, Claude Opus 4.8 made substantial improvements on Click (86.3 to 92.5%) in addition to Drag+Move (41.0 to 74.3%) category over its predecessor Opus 4.7. This indicates the likelihood for purposeful training relating to pointer behavior for Opus 4.8. Claude models are also excellent on Keyboard and Scroll type actions, beating out all other models by a healthy margin. For instance, Claude models score $\geq 89.2\%$ on Keyboard and $\geq 84.1\%$ on Scroll behavior while no other model reaches 80% on either category.

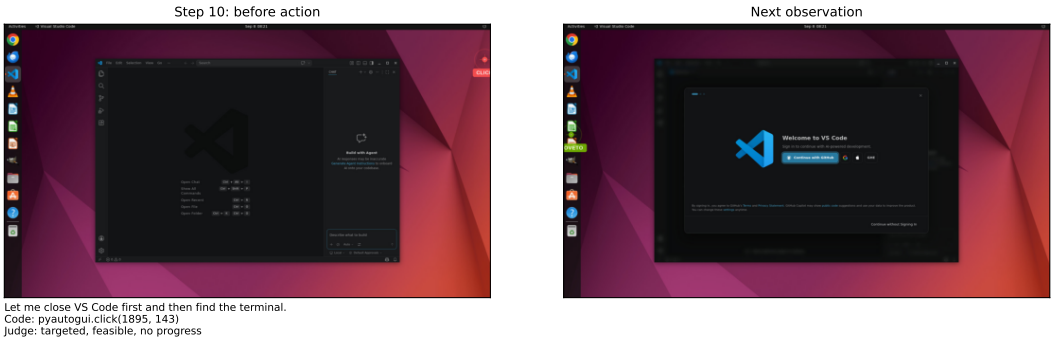


Figure 4: Minimax M3 fails to click on the correct coordinates in order to perform the desired action.

Among tested open-weight models, Qwen 3.8 models stand out despite being much smaller (27B to 125B) compared to other models (428B to 2.8T) as well as their own prior generations of similar sizes (Qwen 3.5 and 3.6), suggesting purposeful training on Qwen 3.8 to do well on GUI manipulation.

Subgoal-Level Efficiency Inspections of OSWorld-Pro trajectories reveal that models can show extremely inefficient behavior despite completing all subgoals, with an example from Claude Opus 5 Max in Fig. 5. To quantify such insights, we consider the average number of steps that the model spends on subgoals of varying status - completed (Successful Effort), feasible but not completed (Failure Persistence) and attempted but not feasible (Infeasible Persistence). Successful Effort represents how directly models complete tasks without detours or unnecessary verification. At this stage, GPT-5.6-Sol and Kimi K3 are efficient at 4.9 and 5.1 steps/subgoal compared to Sonnet 5 and Minimax M3 at 9.9 and 21.2 steps respectively. Failure Persistence represents how hard models try when they fail to eventually complete a feasible subgoal, which broadly correlates well with model efficiency on Successful Effort. Infeasible Persistence represents how quickly models recognize infeasible tasks and give up. Some models like Opus 5 (9.5 steps/subgoal) and GPT-5.6-Terra (12.1) recognize infeasibility rapidly while others persist for more steps (e.g. Sonnet 5 at 39.8).

7 Conclusion

We present OSWorld-Pro, the first process-based evaluation benchmark for Computer-Use Agents (CUAs) to complement outcome-based evaluations such as OSWorld. Based on over 67,000 human-annotations, OSWorld-Pro is a challenging benchmark - especially for open-weight models - that reveals insights into failure modes that have previously eluded outcome-based evaluations (e.g. subgoal-irrelevant behavior and click-based operations). We believe OSWorld-Pro is a critical step to improving CUA evaluation that also comes with potential subsequent applications to guide the performance and efficiency improvement of CUAs (through harness optimization or process reward signals in reinforcement learning), which we leave as future work.

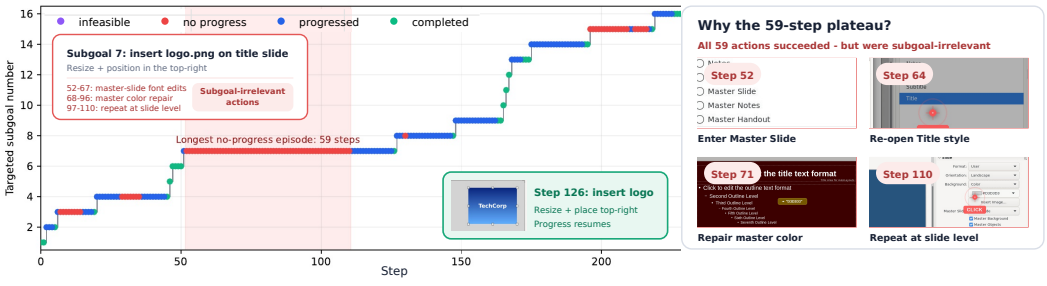


Figure 5: OSWorld-Pro reveals inefficiency of strong models such as Claude Opus 5 Max beyond what outcome-based evaluation alone (e.g. in OSWorld) can show. Opus 5 Max was stuck on multiple subgoals without any progress including once for 59 steps as it performed goal-irrelevant actions.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks:

1. Generate synthetic data sets
2. Implement methods
3. Clean and reformat dataset
4. Support qualitative and thematic data analysis

We have not used generative AI tools for the following tasks:

1. Help develop theoretical models or conceptual frameworks
2. Propose or refine hypotheses
3. Interpret results
4. Design or provide feedback on research methodology or experiments

The remaining disclosure tasks are not applicable to this work:

1. Formulate mathematical claims
2. Provide critical ingredients for proving mathematical claims
3. Assist in the writing of proofs
4. Assist with translation

Additionally, we used generative AI tools for:

1. Create or modify scientific figures or images
2. Create or edit software code

We have reviewed all AI-assisted work.

1. LLM-generated code was verified and tested for correctness by 2 authors
2. Data visualizations were checked by authors against the supplied data to ensure data integrity
3. Human annotators verified synthetic datasets generated, cleaned and reformatted with AI

We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

All data collection carried out on this project was performed by our vendor, following internal reviews on ethical and legal standards prior to the start of the project. All annotators engaged for this project were provided with transparent pay rates before work begins, timely payment on a fixed schedule as well as reasonable working hours and break guidance. If needed, annotators have access to confidential escalation paths for concerns as well as the removal of work content that may create undue risk without additional safeguards. All annotators were paid in accordance to applicable local labor laws as well as internal standards for worker protection and fair compensation.

REPRODUCIBILITY STATEMENT

Procedures for data collection has been extensively documented in §3, B and C. Evaluation details are in §4.1, 5 and E.

REFERENCES

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. Healthbench: Evaluating large language models towards improved human health, 2025. URL https://arxiv.org/abs/2505.08775.

Artificial-Analysis. Gdpval-aa. https://artificialanalysis.ai/evaluations/gdpval-aa, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Mariya Davydova, Daniel Jeffries, Patrick Barker, Arturo Márquez Flores, and Sinéad Ryan. Osuniverse: Benchmark for multimodal gui-navigation ai agents, 2025. URL https://arxiv.org/abs/2505.03570.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyu Hou, Chenhao Xu, Chenze Shao, Chong Ruan, Conner Sun, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Donghao Li, Dongjie Ji, Erhang Li, Fang Wei, Fangyun Lin, Fangzhou Yuan, Feiyu Xia, Fucong Dai, Guangbo Hao, Guanting Chen, Guoai Cao, Guolai Meng, Guowei Li, Han Yu, Han Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoling Zhang, Haoming Luo, Haoran Wei, Haotian Yuan, Haowei Zhang, Haowen Luo, Haoyu Chen, Haozhe Ji, Hengqing Zhang, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, J Yang, JQ Zhu, Jia Luo, Jia Song, Jia Yu, Jialiang Huang, Jialu Cai, Jian Liang, Jiangting Zhou, Jiasheng Ye, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jieyu Yang, Jin Chen, Jin Yan, Jingchang Chen, Jingli Zhou, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jingzi Zhou, Jinhua Zhu, Jiping Yu, Joseph Sun, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junmin Zheng, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Leyi Xia, Li Zhang, Liang Zhao, Lihua Guo, Lingxiao Luo, Linwang Ma, Linyan Zhu, Litong Wang, Liyu Cai, Liyue Zhang, Longhao Chen, MS Di, MY Xu, Max Mei, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Mingxu Zhou, Minmin Han, Ning Wang, Panpan Huang, Panpan Wang, Peixin Cong, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Qiwei Jiang, Rui Tian, Ruifan Xu, Ruijie Lu, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqian Chen, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, Ruyi Chen, SH Liu, Shanghao Lu, Shangmian Sun, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoheng Nie, Shaoqing Wu, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Shuying Yu, Songyang Zhou, Tao Ni, Tao Yun, Tian Jin, Tian Pei, Tian Ye, Tianle Lin, Tianran Ji, Tianyi Cui, Tianyuan Yue, Tingting Yu, Tun Wang, W Zhang, WL Xiao, Wangding Zeng, Wei An, Weilin Zhao, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjing Yao, Wenjun Gao, Wenkai Yang, Wenlve Huang, Wenqing Hou, Wentao Zhang, Wenting Ma, Xi Gao, Xiang He, Xiangwen Wang, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingchen Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyu Zhang, Xu Chen, Xuanyu Wang, Xuecheng Su, Xueyin Chen, Xuheng Lin, Xuwei Fu, YC Yan, YQ Wang, YW Ma, Yanfeng Luo, Yang Zhang, Yanhong Xu, Yanru Ma, Yanwen Huang, Yao Li, Yao Li, Yao Xu, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Shao, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yijia Wu, Yiliang Xiong, Yiling Ma, Ying He, Ying Tang, Ying Zhou, Yingjia Luo, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiang Zhang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, YuKun Li, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuanhao Li, Yuduan Wang, Yuehan Yang, Yuer Xu, Yuhan Wu, Yuhao Meng, Yuheng Zou, Yukun Zha, Yunfan Xiong, Yupeng Chen, Yuping Lin, Yuqian Cao, Yuqian Wang, Yushun Zhang, Yuting Yan, Yutong Lin, Yuxian Gu, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuxuan Zhou, Yuyang Zhou, Yuzhen Huang, ZF Wu, Zehao Wang, Zehua Zhao, Zehui Ren, Zekai Zhang, Zhangli Sha, Zhe Fu, Zhe Ju, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zheren Gao, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhixuan Chen, Zhiyu Wu, Zhizhou Ren, Zhongyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang,

Zihua Qu, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Ziyi Wan, Zizheng Pan, and Zongqing Yao. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains, 2025. URL https://arxiv.org/abs/2507.17746.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code, 2024. URL https://arxiv.org/abs/2403.07974.

Hongrui Jia, Jitong Liao, Xi Zhang, Haiyang Xu, Tianbao Xie, Chaoya Jiang, Ming Yan, Si Liu, Wei Ye, and Fei Huang. Osworld-mcp: Benchmarking mcp tool invocation in computer-use agents, 2025. URL https://arxiv.org/abs/2510.24563.

Jaehun Jung, Ximing Lu, Brandon Cui, Muhammad Khalifa, Shaokun Zhang, Hao Zhang, Jin Xu, Amala Sanjay Deshmukh, Karan Sapra, Andrew Tao, Yejin Choi, Jan Kautz, Mingjie Liu, and Yi Dong. Procua-sft technical report, 2026. URL https://arxiv.org/abs/2606.17321.

Jindong Li, Yali Fu, Li Fan, Jiahong Liu, Yao Shu, Chengwei Qin, Menglin Yang, Irwin King, and Rex Ying. Implicit reasoning in large language models: A comprehensive survey, 2025. URL https://arxiv.org/abs/2509.02350.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

LMSys. Arena-hard-auto leaderboard. https://github.com/lm-sys/arena-hard-auto, 2024.

OpenAI. Openai api pricing. https://developers.openai.com/api/docs/pricing, 2026a.

OpenAI. Huggingface incident and the road ahead. https://openai.com/index/hugging-face-incident-and-the-road-ahead/, 2026b.

OpenRouter. Openrouter. https://openrouter.ai/models?fmt=table, 2025.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra, Adam Khoja, Ryan Kim, Richard Ren, Jason Hausenloy, Oliver Zhang, Mantas Mazeika, Dmitry Dodonov, Tung Nguyen, Jaeho Lee, Daron Anderson, Mikhail Doroshenko, Alun Cennyth Stokes, Mobeen Mahmood, Oleksandr Pokutnyi, Oleg Iskra, Jessica P. Wang, John-Clark Levin, Mstyslav Kazakov, Fiona Feng, Steven Y. Feng, Haoran Zhao, Michael Yu, Varun Gangal, Chelsea Zou, Zihan Wang, Serguei Popov, Robert Gerbicz, Geoff Galgon, Johannes Schmitt, Will Yeadon, Yongki Lee, Scott Sauers, Alvaro Sanchez, Fabian Giska, Marc Roth, Søren Riis, Saiteja Utpala, Noah Burns, Gashaw M. Goshu, Mohinder Maheshbhai Naiya, Chidozie Agu, Zachary Giboney, Antrell Cheatom, Francesco Fournier-Facio, Sarah-Jane Crowson, Lennart Finke, Zerui Cheng, Jennifer Zampese, Ryan G. Hoerr, Mark Nandor, Hyunwoo Park, Tim Gehrunger, Jiaqi Cai, Ben McCarty, Alexis C Garretson, Edwin Taylor, Damien Sileo, Qiuyu Ren, Usman Qazi, Lianghui Li, Jungbae Nam, John B. Wydallis, Pavel Arkhipov, Jack Wei Lun Shi, Aras Bacho, Chris G. Willcocks, Hangrui Cao, Sumeet Motwani, Emily de Oliveira Santos, Johannes Veith, Edward Vendrow, Doru Cojoc, Kengo Zenitani, Joshua Robinson, Longke Tang, Yuqi Li, Joshua Vendrow, Natanael Wildner Fraga, Vladyslav Kuchkin, Andrey Pupasov Maksimov, Pierre Marion, Denis Efremov, Jayson Lynch, Kaiqu Liang, Aleksandar Mikov, Andrew Gritsevskiy, Julien Guillod, Gözdenur Demir, Dakotah Martinez, Ben Pageler, Kevin Zhou, Saeed Soori, Ori Press, Henry Tang, Paolo Rissone, Sean R. Green, Lina Brüssel, Moon Twayana, Aymeric Dieuleveut, Joseph Marvin Imperial, Ameya Prabhu, Jinzhou Yang, Nick Crispino, Arun Rao, Dimitri Zvonkine, Gabriel Loiseau, Mikhail Kalinin, Marco Lukas, Ciprian Manolescu, Nate Stambaugh, Subrata Mishra, Tad Hogg, Carlo Bosio, Brian P Coppola, Julian Salazar, Jaehyeok Jin, Rafael Sayous, Stefan Ivanov,

Philippe Schwaller, Shaipranesh Senthilkuma, Andres M Bran, Andres Algaba, Kelsey Van den Houte, Lynn Van Der Sypt, Brecht Verbeken, David Noever, Alexei Kopylov, Benjamin Myklebust, Bikun Li, Lisa Schut, Evgenii Zheltonozhskii, Qiaochu Yuan, Derek Lim, Richard Stanley, Tong Yang, John Maar, Julian Wykowski, Martí Oller, Anmol Sahu, Cesare Giulio Ardito, Yuzheng Hu, Ariel Ghislain Kemogne Kamdoum, Alvin Jin, Tobias Garcia Vilchis, Yuexuan Zu, Martin Lackner, James Koppel, Gongbo Sun, Daniil S. Antonenko, Steffi Chern, Bingchen Zhao, Pierrot Arsene, Joseph M Cavanagh, Daofeng Li, Jiawei Shen, Donato Crisostomi, Wenjin Zhang, Ali Dehghan, Sergey Ivanov, David Perrella, Nurdin Kaparov, Allen Zang, Ilia Sucholutsky, Arina Kharlamova, Daniil Orel, Vladislav Poritski, Shalev Ben-David, Zachary Berger, Parker Whitfill, Michael Foster, Daniel Munro, Linh Ho, Shankar Sivarajan, Dan Bar Hava, Aleksey Kuchkin, David Holmes, Alexandra Rodriguez-Romero, Frank Sommerhage, Anji Zhang, Richard Moat, Keith Schneider, Zakayo Kazibwe, Don Clarke, Dae Hyun Kim, Felipe Meneguitti Dias, Sara Fish, Veit Elser, Tobias Kreiman, Victor Efren Guadarrama Vilchis, Immo Klose, Ujjwala Anantheswaran, Adam Zweiger, Kaivalya Rawal, Jeffery Li, Jeremy Nguyen, Nicolas Daans, Haline Heidinger, Maksim Radionov, Václav Rozhoň, Vincent Ginis, Christian Stump, Niv Cohen, Rafał Poświata, Josef Tkadlec, Alan Goldfarb, Chenguang Wang, Piotr Padlewski, Stanislaw Barzowski, Kyle Montgomery, Ryan Stendall, Jamie Tucker-Foltz, Jack Stade, T. Ryan Rogers, Tom Goertzen, Declan Grabb, Abhishek Shukla, Alan Givré, John Arnold Ambay, Archan Sen, Muhammad Fayez Aziz, Mark H Inlow, Hao He, Ling Zhang, Younesse Kaddar, Ivar Ängquist, Yanxu Chen, Harrison K Wang, Kalyan Ramakrishnan, Elliott Thornley, Antonio Terpin, Hailey Schoelkopf, Eric Zheng, Avishy Carmi, Ethan D. L. Brown, Kelin Zhu, Max Bartolo, Richard Wheeler, Martin Stehberger, Peter Bradshaw, JP Heimonen, Kaustubh Sridhar, Ido Akov, Jennifer Sandlin, Yury Makarychev, Joanna Tam, Hieu Hoang, David M. Cunningham, Vladimir Goryachev, Demosthenes Patramanis, Michael Krause, Andrew Redenti, David Aldous, Jesyin Lai, Shannon Coleman, Jiangnan Xu, Sangwon Lee, Ilias Magoulas, Sandy Zhao, Ning Tang, Michael K. Cohen, Orr Paradise, Jan Hendrik Kirchner, Maksym Ovchynnikov, Jason O. Matos, Adithya Shenoy, Michael Wang, Yuzhou Nie, Anna Sztyber-Betley, Paolo Faraboschi, Robin Riblet, Jonathan Crozier, Shiv Halasyamani, Shreyas Verma, Prashant Joshi, Eli Meril, Ziqiao Ma, Jérémy Andréoletti, Raghav Singhal, Jacob Platnick, Volodymyr Nevirkovets, Luke Basler, Alexander Ivanov, Seri Khoury, Nils Gustafsson, Marco Piccardo, Hamid Mostaghimi, Qijia Chen, Virendra Singh, Tran Quoc Khánh, Paul Rosu, Hannah Szlyk, Zachary Brown, Himanshu Narayan, Aline Menezes, Jonathan Roberts, William Alley, Kunyang Sun, Arkil Patel, Max Lamparth, Anka Reuel, Linwei Xin, Hanmeng Xu, Jacob Loader, Freddie Martin, Zixuan Wang, Andrea Achilleos, Thomas Preu, Tomek Korbak, Ida Bosio, Fereshteh Kazemi, Ziye Chen, Biró Bálint, Eve J. Y. Lo, Jiaqi Wang, Maria Inês S. Nunes, Jeremiah Milbauer, M Saiful Bari, Zihao Wang, Behzad Ansarinejad, Yewen Sun, Stephane Durand, Hossam Elgnainy, Guillaume Douville, Daniel Tordera, George Balabanian, Hew Wolff, Lynna Kvistad, Hsiaoyun Milliron, Ahmad Sakor, Murat Eron, Andrew Favre D. O., Shailesh Shah, Xiaoxiang Zhou, Firuz Kamalov, Sherwin Abdoli, Tim Santens, Shaul Barkan, Allison Tee, Robin Zhang, Alessandro Tomasiello, G. Bruno De Luca, Shi-Zhuo Looi, Vinh-Kha Le, Noam Kolt, Jiayi Pan, Emma Rodman, Jacob Drori, Carl J Fossum, Niklas Muennighoff, Milind Jagota, Ronak Pradeep, Honglu Fan, Jonathan Eicher, Michael Chen, Kushal Thaman, William Merrill, Moritz Firsching, Carter Harris, Stefan Ciobâcă, Jason Gross, Rohan Pandey, Ilya Gusev, Adam Jones, Shashank Agnihotri, Pavel Zhelnov, Mohammadreza Mofayezi, Alexander Piperski, David K. Zhang, Kostiantyn Dobarskyi, Roman Leventov, Ignat Soroko, Joshua Duersch, Vage Taamazyan, Andrew Ho, Wenjie Ma, William Held, Ruicheng Xian, Armel Randy Zebaze, Mohanad Mohamed, Julian Noah Leser, Michelle X Yuan, Laila Yacar, Johannes Lengler, Katarzyna Olszewska, Claudio Di Fratta, Edson Oliveira, Joseph W. Jackson, Andy Zou, Muthu Chidambaram, Timothy Manik, Hector Haffenden, Dashiell Stander, Ali Dasouqi, Alexander Shen, Bita Golshani, David Stap, Egor Kretov, Mikalai Uzhou, Alina Borisovna Zhidkovskaya, Nick Winter, Miguel Orbegozo Rodriguez, Robert Lauff, Dustin Wehr, Colin Tang, Zaki Hossain, Shaun Phillips, Fortuna Samuele, Fredrik Ekström, Angela Hammon, Oam Patel, Faraz Farhidi, George Medley, Forough Mohammadzadeh, Madellene Peñaflor, Haile Kassahun, Alena Friedrich, Rayner Hernandez Perez, Daniel Pyda, Taom Sakal, Omkar Dhamane, Ali Khajegili Mirabadi, Eric Hallman, Kenchi Okutsu, Mike Battaglia, Mohammad Maghsoudimehrabani, Alon Amit, Dave Hulbert, Roberto Pereira, Simon Weber, Handoko, Anton Peristyy, Stephen Malina, Mustafa Mehkary, Rami Aly, Frank Reidegeld, Anna-Katharina Dick, Cary Friday, Mukhwinder Singh, Hassan Shapourian, Wanyoung Kim, Mariana Costa, Hubeyb Gurdogan, Harsh Kumar, Chiara Ceconello, Chao Zhuang, Haon Park, Micah Carroll, Andrew R. Tawfeek, Stefan Steinerberger, Daattavya Aggarwal, Michael Kirchhof, Linjie Dai, Evan Kim, Johan Ferret, Jainam Shah, Yuzhou Wang, Minghao Yan, Krzysztof Burdzy, Lixin

Zhang, Antonio Franca, Diana T. Pham, Kang Yong Loh, Joshua Robinson, Abram Jackson, Paolo Giordano, Philipp Petersen, Adrian Cosma, Jesus Colino, Colin White, Jacob Votava, Vladimir Vinnikov, Ethan Delaney, Petr Spelda, Vit Stritecky, Syed M. Shahid, Jean-Christophe Mourrat, Lavr Vetoshkin, Koen Sponselee, Renas Bacho, Zheng-Xin Yong, Florencia de la Rosa, Nathan Cho, Xiuyu Li, Guillaume Malod, Orion Weller, Guglielmo Albani, Leon Lang, Julien Laurendeau, Dmitry Kazakov, Fatimah Adesanya, Julien Portier, Lawrence Hollom, Victor Souza, Yuchen Anna Zhou, Julien Degorre, Yiğit Yalın, Gbenga Daniel Obikoya, Rai, Filippo Bigi, M. C. Boscá, Oleg Shumar, Kaniuar Bacho, Gabriel Recchia, Mara Popescu, Nikita Shulga, Ngefor Mildred Tanwie, Thomas C. H. Lux, Ben Rank, Colin Ni, Matthew Brooks, Alesia Yakimchyk, Huanxu, Liu, Stefano Cavalleri, Olle Häggström, Emil Verkama, Joshua Newbould, Hans Gundlach, Leonor Brito-Santana, Brian Amaro, Vivek Vajipey, Rynaa Grover, Ting Wang, Yosi Kratish, Wen-Ding Li, Sivakanth Gopi, Andrea Caciolai, Christian Schroeder de Witt, Pablo Hernández-Cámara, Emanuele Rodolà, Jules Robins, Dominic Williamson, Vincent Cheng, Brad Raynor, Hao Qi, Ben Segev, Jingxuan Fan, Sarah Martinson, Erik Y. Wang, Kaylie Hausknecht, Michael P. Brenner, Mao Mao, Christoph Demian, Peyman Kassani, Xinyu Zhang, David Avagian, Eshawn Jessica Scipio, Alon Ragoler, Justin Tan, Blake Sims, Rebeka Plecnik, Aaron Kirtland, Omer Faruk Bodur, D. P. Shinde, Yan Carlos Leyva Labrador, Zahra Adoul, Mohamed Zekry, Ali Karakoc, Tania C. B. Santos, Samir Shamseldeen, Loukmane Karim, Anna Liakhovitskaia, Nate Resman, Nicholas Farina, Juan Carlos Gonzalez, Gabe Maayan, Earth Anderson, Rodrigo De Oliveira Pena, Elizabeth Kelley, Hodjat Mariji, Rasoul Pouriamanesh, Wentao Wu, Ross Finocchio, Ismail Alarab, Joshua Cole, Danyelle Ferreira, Bryan Johnson, Mohammad Safdari, Liangti Dai, Siriphan Arthornthurasuk, Isaac C. McAlister, Alejandro José Moyano, Alexey Pronin, Jing Fan, Angel Ramirez-Trinidad, Yana Malysheva, Daphiny Pottmaier, Omid Taheri, Stanley Stepanic, Samuel Perry, Luke Askew, Raúl Adrián Huerta Rodríguez, Ali M. R. Minissi, Ricardo Lorena, Krishnamurthy Iyer, Arshad Anil Fasiludeen, Ronald Clark, Josh Ducey, Matheus Piza, Maja Somrak, Eric Vergo, Juehang Qin, Benjámin Borbás, Eric Chu, Jack Lindsey, Antoine Jallon, I. M. J. McInnis, Evan Chen, Avi Semler, Luk Gloor, Tej Shah, Marc Carauleanu, Pascal Lauer, Tran uc Huy, Hossein Shahrtash, Emilien Duc, Lukas Lewark, Assaf Brown, Samuel Albanie, Brian Weber, Warren S. Vaz, Pierre Clavier, Yiyang Fan, Gabriel Poesia Reis e Silva, Long, Lian, Marcus Abramovitch, Xi Jiang, Sandra Mendoza, Murat Islam, Juan Gonzalez, Vasilios Mavroudis, Justin Xu, Pawan Kumar, Laxman Prasad Goswami, Daniel Bugas, Nasser Heydari, Ferenc Jeanplong, Thorben Jansen, Antonella Pinto, Archimedes Apronti, Abdallah Galal, Ng Ze-An, Ankit Singh, Tong Jiang, Joan of Arc Xavier, Kanu Priya Agarwal, Mohammed Berkani, Gang Zhang, Zhehang Du, Benedito Alves de Oliveira Junior, Dmitry Malishev, Nicolas Remy, Taylor D. Hartman, Tim Tarver, Stephen Mensah, Gautier Abou Loume, Wiktor Morak, Farzad Habibi, Sarah Hoback, Will Cai, Javier Gimenez, Roselynn Grace Montecillo, Jakub Łucki, Russell Campbell, Asankhaya Sharma, Khalida Meer, Shreen Gul, Daniel Espinosa Gonzalez, Xavier Alapont, Alex Hoover, Gunjan Chhablani, Freddie Vargus, Arunim Agarwal, Yibo Jiang, Deepakkumar Patil, David Outevsky, Kevin Joseph Scaria, Rajat Maheshwari, Abdelkader Dendane, Priti Shukla, Ashley Cartwright, Sergei Bogdanov, Niels Mündler, Sören Möller, Luca Arnaboldi, Kunvar Thaman, Muhammad Rehan Siddiqi, Prajvi Saxena, Himanshu Gupta, Tony Fruhauff, Glen Sherman, Mátyás Vincze, Siranut Usawasutsakorn, Dylan Ler, Anil Radhakrishnan, Innocent Enyekwe, Sk Md Salauddin, Jiang Muzhen, Aleksandr Maksapetyan, Vivien Rossbach, Chris Harjadi, Mohsen Bahaloohoreh, Claire Sparrow, Jasdeep Sidhu, Sam Ali, Song Bian, John Lai, Eric Singer, Justine Leon Uro, Greg Bateman, Mohamed Sayed, Ahmed Menshawy, Darling Duclosel, Dario Bezzi, Yashaswini Jain, Ashley Aaron, Murat Tiryakioglu, Sheeshram Siddh, Keith Krenek, Imad Ali Shah, Jun Jin, Scott Creighton, Denis Peskoff, Zienab EL-Wasif, Ragavendran P V, Michael Richmond, Joseph McGowan, Tejal Patwardhan, Hao-Yu Sun, Ting Sun, Nikola Zubić, Samuele Sala, Stephen Ebert, Jean Kaddour, Manuel Schottdorf, Dianzhuo Wang, Gerol Petruzella, Alex Meiburg, Tilen Medved, Ali ElSheikh, S Ashwin Hebbar, Lorenzo Vaquero, Xianjun Yang, Jason Poulos, Vilém Zouhar, Sergey Bogdanik, Mingfang Zhang, Jorge Sanz-Ros, David Anugraha, Yinwei Dai, Anh N. Nhu, Xue Wang, Ali Anil Demircali, Zhibai Jia, Yuyin Zhou, Juncheng Wu, Mike He, Nitin Chandok, Aarush Sinha, Gaoxiang Luo, Long Le, Mickaël Noyé, Michał Perełkiewicz, Ioannis Pantidis, Tianbo Qi, Soham Sachin Purohit, Letitia Parcalabescu, Thai-Hoa Nguyen, Genta Indra Winata, Edoardo M. Ponti, Hanchen Li, Kaustubh Dhole, Jongee Park, Dario Abbondanza, Yuanli Wang, Anupam Nayak, Diogo M. Caetano, Antonio A. W. L. Wong, Maria del Rio-Chanona, Dániel Kondor, Pieter Francois, Ed Chalstrey, Jakob Zsambok, Dan Hoyer, Jenny Reddish, Jakob Hauser, Francisco-Javier Rodrigo-Ginés, Suchandra Datta, Maxwell Shepherd, Thom Kamphuis, Qizheng Zhang, Hyunjun Kim, Ruiji Sun, Jianzhu Yao, Franck Dernoncourt, Satyapriya Krishna, Sina

Rismanchian, Bonan Pu, Francesco Pinto, Yingheng Wang, Kumar Shridhar, Kalon J. Overholt, Glib Briia, Hieu Nguyen, David, Soler Bartomeu, Tony CY Pang, Adam Wecker, Yifan Xiong, Fanfei Li, Lukas S. Huber, Joshua Jaeger, Romano De Maddalena, Xing Han Lù, Yuhui Zhang, Claas Beger, Patrick Tser Jern Kon, Sean Li, Vivek Sanker, Ming Yin, Yihao Liang, Xinlu Zhang, Ankit Agrawal, Li S. Yifei, Zechen Zhang, Mu Cai, Yasin Sonmez, Costin Cozianu, Changhao Li, Alex Slen, Shoubin Yu, Hyun Kyu Park, Gabriele Sarti, Marcin Briański, Alessandro Stolfo, Truong An Nguyen, Mike Zhang, Yotam Perlitz, Jose Hernandez-Orallo, Runjia Li, Amin Shabani, Felix Juefei-Xu, Shikhar Dhingra, Orr Zohar, My Chiffon Nguyen, Alexander Pondaven, Abdurrahim Yilmaz, Xuandong Zhao, Chuanyang Jin, Muyan Jiang, Stefan Todoran, Xinyao Han, Jules Kreuer, Brian Rabern, Anna Plassart, Martino Maggetti, Luther Yap, Robert Geirhos, Jonathon Kean, Dingsu Wang, Sina Mollaei, Chenkai Sun, Yifan Yin, Shiqi Wang, Rui Li, Yaowen Chang, Anjiang Wei, Alice Bizeul, Xiaohan Wang, Alexandre Oliveira Arrais, Kushin Mukherjee, Jorge Chamorro-Padial, Jiachen Liu, Xingyu Qu, Junyi Guan, Adam Bouyamourn, Shuyu Wu, Martyna Plomecka, Junda Chen, Mengze Tang, Jiaqi Deng, Shreyas Subramanian, Haocheng Xi, Haoxuan Chen, Weizhi Zhang, Yinuo Ren, Haoqin Tu, Sejong Kim, Yushun Chen, Sara Vera Marjanović, Junwoo Ha, Grzegorz Luczyna, Jeff J. Ma, Zewen Shen, Dawn Song, Cedegao E. Zhang, Zhun Wang, Gaël Gendron, Yunze Xiao, Leo Smucker, Erica Weng, Kwok Hao Lee, Zhe Ye, Stefano Ermon, Ignacio D. Lopez-Miguel, Theo Knights, Anthony Gitter, Namkyu Park, Boyi Wei, Hongzheng Chen, Kunal Pai, Ahmed Elkhanany, Han Lin, Philipp D. Siedler, Jichao Fang, Ritwik Mishra, Károly Zsolnai-Fehér, Xilin Jiang, Shadab Khan, Jun Yuan, Rishab Kumar Jain, Xi Lin, Mike Peterson, Zhe Wang, Aditya Malusare, Maosen Tang, Isha Gupta, Ivan Fosin, Timothy Kang, Barbara Dworakowska, Kazuki Matsumoto, Guangyao Zheng, Gerben Sewuster, Jorge Pretel Villanueva, Ivan Rannev, Igor Chernyavsky, Jiale Chen, Deepayan Banik, Ben Racz, Wenchao Dong, Jianxin Wang, Laila Bashmal, Duarte V. Gonçalves, Wei Hu, Kaushik Bar, Ondrej Bohdal, Atharv Singh Patlan, Shehzaad Dhuliawala, Caroline Geirhos, Julien Wist, Yuval Kansal, Bingsen Chen, Kutay Tire, Atak Talay Yücel, Brandon Christof, Veerupaksh Singla, Zijian Song, Sanxing Chen, Jiaxin Ge, Kaustubh Ponkshe, Isaac Park, Tianneng Shi, Martin Q. Ma, Joshua Mak, Sherwin Lai, Antoine Moulin, Zhuo Cheng, Zhanda Zhu, Ziyi Zhang, Vaidehi Patil, Ketan Jha, Qiutong Men, Jiaxuan Wu, Tianchi Zhang, Bruno Hebling Vieira, Alham Fikri Aji, Jae-Won Chung, Mohammed Mahfoud, Ha Thi Hoang, Marc Sperzel, Wei Hao, Kristof Meding, Sihan Xu, Vassilis Kostakos, Davide Manini, Yueying Liu, Christopher Toukmaji, Jay Paek, Eunmi Yu, Arif Engin Demircali, Zhiyi Sun, Ivan Dewerpe, Hongsen Qin, Roman Pflugfelder, James Bailey, Johnathan Morris, Ville Heilala, Sybille Rosset, Zishun Yu, Peter E. Chen, Woongyeong Yeo, Eeshaan Jain, Ryan Yang, Sreekar Chigurupati, Julia Chernyavsky, Sai Prajwal Reddy, Subhashini Venugopalan, Hunar Batra, Core Francisco Park, Hieu Tran, Guilherme Maximiano, Genghan Zhang, Yizhuo Liang, Hu Shiyu, Rongwu Xu, Rui Pan, Siddharth Suresh, Ziqi Liu, Samaksh Gulati, Songyang Zhang, Peter Turchin, Christopher W. Bartlett, Christopher R. Scotese, Phuong M. Cao, Aakaash Nattanmai, Gordon McKellips, Anish Cheraku, Asim Suhail, Ethan Luo, Marvin Deng, Jason Luo, Ashley Zhang, Kavin Jindel, Jay Paek, Kasper Halevy, Allen Baranov, Michael Liu, Advaith Avadhanam, David Zhang, Vincent Cheng, Brad Ma, Evan Fu, Liam Do, Joshua Lass, Hubert Yang, Surya Sunkari, Vishruth Bharath, Violet Ai, James Leung, Rishit Agrawal, Alan Zhou, Kevin Chen, Tejas Kalpathi, Ziqi Xu, Gavin Wang, Tyler Xiao, Erik Maung, Sam Lee, Ryan Yang, Roy Yue, Ben Zhao, Julia Yoon, Sunny Sun, Aryan Singh, Ethan Luo, Clark Peng, Tyler Osbey, Taozhi Wang, Daryl Echeazu, Hubert Yang, Timothy Wu, Spandan Patel, Vidhi Kulkarni, Vijaykaarti Sundarapandiyan, Ashley Zhang, Andrew Le, Zafir Nasim, Srikar Yalam, Ritesh Kasamsetty, Soham Samal, Hubert Yang, David Sun, Nihar Shah, Abhijeet Saha, Alex Zhang, Leon Nguyen, Laasya Nagumalli, Kaixin Wang, Alan Zhou, Aidan Wu, Jason Luo, Anwith Telluri, Summer Yue, Alexandr Wang, and Dan Hendrycks. Humanity's last exam, 2025. URL https://arxiv.org/abs/2501.14249.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Toyama, Robert Berry, Divya Tyamagundlu, Timothy Lillicrap, and Oriana Riva. Androidworld: A dynamic benchmarking environment for autonomous agents, 2025. URL https://arxiv.org/abs/2405.14573.

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 qa benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

Vincent Siu, Manasi Sharma, Dawn Song, Daniel Yue Zhang, Chenguang Wang, and Ying Liu. Chainworld: Composing long-horizon desktop workloads from atomic osworld tasks, 2026. URL https://arxiv.org/abs/2606.21654.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. Paperbench: Evaluating ai's ability to replicate ai research, 2025. URL https://arxiv.org/abs/2504.01848.

Xinyuan Wang, Bowen Wang, Dunjie Lu, Junlin Yang, Tianbao Xie, Junli Wang, Jiaqi Deng, Xiaole Guo, Yiheng Xu, Chen Henry Wu, Zhennan Shen, Zhuokai Li, Ryan Li, Xiaochuan Li, Junda Chen, Zheng Boyuan, LI PEIHANG, Fangyu Lei, Ruisheng Cao, Yeqiao Fu, Dongchan Shin, Martin Shin, Hu Jiarui, Yuyan Wang, Jixuan Chen, Yuxiao Ye, Danyang Zhang, Yipu Wang, Heng Wang, Diyi Yang, Victor Zhong, Y. Charles, Zhilin Yang, and Tao Yu. OpenCUA: Open foundations for computer-use agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=6iRZvJiC9Q.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024. URL https://arxiv.org/abs/2406.01574.

Zhilin Wang, Jiaqi Zeng, Olivier Delalleau, Daniel Egert, Ellie Evans, Hoo-Chang Shin, Felipe Soares, Yi Dong, and Oleksii Kuchaiev. HelpSteer3: Human-annotated feedback and edit data to empower inference-time scaling in open-ended general-domain tasks. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25640–25662, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1246. URL https://aclanthology.org/2025.acl-long.1246/.

Zhilin Wang, Jaehun Jung, Ximing Lu, Shizhe Diao, Ellie Evans, Jiaqi Zeng, Pavlo Molchanov, Yejin Choi, Jan Kautz, and Yi Dong. Profbench: Multi-domain rubrics requiring professional knowledge to answer and judge. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=VwNzKPqBxk.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha Venkat Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. Livebench: A challenging, contamination-free LLM benchmark. In *The Thirteenth International Conference on Learning Representations*, 2025.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=tN61DTr4Ed.

XLANG-Lab. Osworld github. https://github.com/xlang-ai/osworld, 2024.

XLANG-Lab. Osworld website. https://osworld-v1.xlang.ai/, 2025.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. ProcessBench: Identifying process errors in mathematical reasoning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1009–1024, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.50. URL https://aclanthology.org/2025.acl-long.50/.

OSWorld-Pro Diversity Example

Rare App: Kid3 - audio tagger for music metadata (not seen in OSWorld)
Goal: Install the Kid3 audio tagging application from the KDE website and use it to edit the metadata for Zhou Xuan - Nights in Shanghai.mp3 in the Music folder, adding the album name and release year.

Subgoals: *[App used]*

1. Download the Kid3 application from the KDE website. *[Chrome]*
2. Install the Kid3 application. *[Terminal]*
3. Open Zhou Xuan - Nights in Shanghai.mp3 from the Music folder in Kid3. *[Kid3]*
4. Add the album name to the metadata of the MP3 file. *[Kid3]*
5. Add the release year to the metadata of the MP3 file. *[Kid3]*

Steps: (e.g. Step 1)

Action:
`pyautogui.click(0.500,0.556)`

Reasoning:
... I can see the Kid3 website is already open in one of the tabs. Let me click on that tab to see the full page and find download instructions

Human Annotation:

Subgoal(s) targeted: **{1}**
Subgoal(s) feasibility: ✓
Subgoal(s) progression: ✓
Subgoal(s) completion: ✗

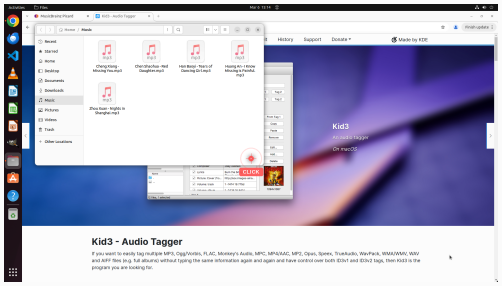


Figure 6: OSWorld-Pro Diversity Example.

A EXAMPLE DATA

OSWorld-Pro Robustness Example

Environment: Oracle Linux 9 with GNOME

Goal: Create a comprehensive quarterly business review package in the server folder consisting of: (1) a LibreOffice Calc workbook named 'Q1_Financials.xlsx' with multi-sheet revenue and expense data, calculated totals using formulas, and a formatted bar chart comparing categories; (2) a LibreOffice Writer report named 'Q1_Executive_Summary.docx' with styled headings, an auto-generated table of contents, body paragraphs, and a summary table referencing the spreadsheet data; and (3) a LibreOffice Impress presentation named 'Q1_Review.pptx' with at least five slides including a title page, agenda, financial overview slide incorporating the chart from Calc, and a recommendations slide, all using consistent visual styling and branding.

Subgoals: *[App used]*

1. Create a LibreOffice Calc workbook named 'Q1_Financials.xlsx' in the server folder containing multi-sheet revenue and expense data. *[LibreOffice Calc]*
2. Add calculated totals using formulas to the 'Q1_Financials.xlsx' workbook. *[LibreOffice Calc]*
3. Insert a formatted bar chart comparing revenue and expense categories into the 'Q1_Financials.xlsx' workbook *[LibreOffice Calc]*
4. Create a LibreOffice Writer document named 'Q1_Executive_Summary.docx' in the server folder containing body paragraphs. *[LibreOffice Writer]*
5. Apply styled headings to the 'Q1_Executive_Summary.docx' document *[LibreOffice Writer]*
6. Insert an auto-generated table of contents into the 'Q1_Executive_Summary.docx' document *[LibreOffice Writer]*
7. Insert a summary table referencing the 'Q1_Financials.xlsx' spreadsheet data into the 'Q1_Executive_Summary.docx' document *[LibreOffice Writer]*
8. Create a LibreOffice Impress presentation named 'Q1_Review.pptx' in the server folder containing a title page slide *[LibreOffice Impress]*
9. Add an agenda slide to the 'Q1_Review.pptx' presentation *[LibreOffice Impress]*
10. Add a financial overview slide incorporating the bar chart from the 'Q1_Financials.xlsx' workbook to the 'Q1_Review.pptx' presentation *[LibreOffice Impress]*
11. Add a recommendations slide to the 'Q1_Review.pptx' presentation *[LibreOffice Impress]*
12. Add at least one additional slide to the 'Q1_Review.pptx' presentation to satisfy the minimum five-slide requirement *[LibreOffice Impress]*
13. Apply consistent visual styling and branding to all slides in the 'Q1_Review.pptx' presentation *[LibreOffice Impress]*

Steps: (e.g. Step 1)

Action:
`pyautogui.doubleClick(0.254,0.104)`

Reasoning:
... Current screen: file manager window showing home directory with Desktop and server folders. I need to open the server folder.

Human Annotation:

Subgoal(s) targeted: **{1}**
Subgoal(s) feasibility: ✓
Subgoal(s) progression: ✓
Subgoal(s) completion: ✗

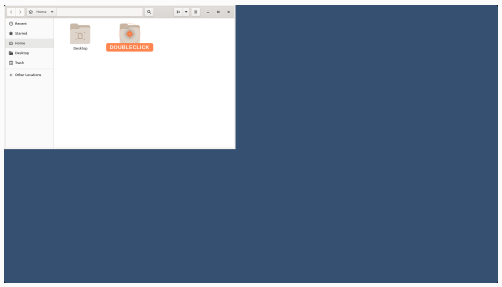


Figure 7: OSWorld-Pro Robustness Example.

B Further Descriptive Statistics

Goals OSWorld-Pro contains 305 unique tasks, with task goals generally at around a few sentences long with an average length of 553.2 characters (std of 262.5, min of 87 and max of 999). As shown in §A, task goals usually contain multiple related objectives for the agent to achieve.

Subgoals The overall goal is decomposed into individual subgoals that can be monitored with more granularity. Specifically, each goal is decomposed into an average of 9.2 subgoals (std of 4.7, min of 2 and max of 27). Subgoals are typically a single concise sentence with 60.2 characters (std of 35.6, min of 8 and max of 327).

Agent Trajectories contain an average of 55.1 steps (std of 33.4, min of 5 and max of 149). Each step contains a screenshot (at 1920 *1080 resolution), a reasoning trace, a natural language description

of the action and a code action. The code action refers to a pyautogui snippet that manipulates the state of the computer such as a click action or keyboard entry. Among all actions, most are in the click category (51.0% click, 3.6% doubleClick, 2.8% rightClick, 2.7% tripleClick), followed by the keyboard category (16.6% press (e.g. holding ctrl+c), 13.6% typewrite, 0.7% write), scrolling category (3.3% scroll, 0.1% hscroll), execution-control category (1.9% sleep, 1.3% terminate, 0.1% wait) and other pointer (2.3% moveTo). For judging purposes by both human judges and LLM-judges, we visualize the code action on top of the screenshot (as a semi-transparent overlay) as some of these actions (e.g. click(125, 61), which are the absolute x,y coordinates) can be hard to interpret without it.

Human Annotations Most steps in agent trajectories only targeted a single subgoal (98.3%) while 1.1, 0.4 and 0.2% target 2, 3 and 4 subgoals respectively. 84.7% of subgoals targeted at these steps were deemed feasible, 59.8% of steps led to a progression in subgoal(s) while 10.6% of steps led to a completion of subgoal(s). Across all steps within an agent trajectory, 23.1% of subgoals were unattempted, 2.0% were attempted but not feasible, 1.1% were feasible but not progressed, 8.4% were progressed but not completed and 65.3% were completed. Subgoals that were attempted but infeasible have an average of 18.8 steps, feasible but not progressed subgoals have 10.5 steps, progressed but not completed subgoals have 18.7 steps and completed subgoals have 6.2 steps.

C Annotator Recruitment

Annotator Countries We recruit annotators from diverse geographic backgrounds to capture a range of computer-use habits and practices.

1. India: 56%
2. Nigeria: 24 %
3. Brazil: 8%
4. Pakistan: 8%
5. Ethiopia: 4%

Technical Pool: Our vendor generally sought for annotators with Engineering-focused education at the Bachelors' level (or above) with hands-on coding experience. Specifically, 55% hold a Bachelor of Engineering, 18% Bachelor of Science and 9% hold a Bachelor of Business Administration, Master of Business Administration, and Post Graduate Degree in Management respectively.

Non-Technical Pool: Our vendor generally sought for annotators with a Bachelor's Degree (or higher) with strong literacy relating to technology. 40% hold a Bachelor of Engineering, 27% Master of Science, 14% Bachelor of Commerce and 7% hold a Bachelor of Science, Bachelor of Design and Master of Business Administration respectively (Percentages do not necessarily add to 100% due to rounding errors).

D Annotation Guidelines

(This appendix reproduces the document given to annotators, with small terminology adjustment to the main text and data fields not used in this paper were removed to avoid confusion)

This set of guidelines defines how to evaluate a trajectory at the goal, subgoal, overall-safety, and per-step levels. It also defines how Reviewer independently score and subsequently compare their judgments with the Annotator's work.

The rubric is designed to separate three questions that must not be conflated:

1. Is the task goal valid enough to evaluate?
2. Is the subgoal decomposition correct and usable?
3. At each step, what did the model target, feasibly attempt, advance, and complete?

D.1 TERMINOLOGY

Term	Meaning
Annotator	The person who produces the primary annotation.
Reviewer	Performs an independent blind pass, then reviews the Annotator's work.
Subgoal	A separately checkable milestone that contributes to the overall goal.
App Used	The primary application used to accomplish a particular subgoal.
Step	One action unit in the trajectory.

D.2 REQUIRED EVALUATION ORDER

Evaluation sequence

1. Score *Goal Clarity, Specificity and Safety*. Continue only if the score is 5 out of 5.
2. Review the subgoal list, one-to-one *App Used* mapping, and sequential dependency. Correct them if needed.
3. For every step, record Targeted Subgoal(s), Feasibility, Progression, and Completion in that order.

D.3 TASK FILTERING

Evaluate whether the goal is clearly defined, unambiguous, and unlikely to induce actions carrying safety, policy, or ethical risks.

Label	Definition
5 Excellent	The goal is explicit, precise, and fully specifies the intended outcome.
4 Good	The goal is clear with minor high-level phrasing.
3 Acceptable	The goal is understandable but vague or underspecified.
2 Poor	The goal is unclear, generic, or partially ambiguous.
1 Very Poor	The goal is contradictory, incomprehensible, or likely to induce safety risks.

Proceed/skip gate

Score 5: proceed with task evaluation.
Score 4 or below: provide the reason and skip the task.

D.4 SUBGOAL LIST AND APP USED CORRECTNESS (BOOLEAN)

The task includes an LLM-generated subgoal list and a corresponding *App Used* list. Review and, when necessary, correct both lists.

Correctness requirements

- **Goal aligned:** every subgoal contributes to the original goal; there are no illogical, unrelated, hallucinated, or out-of-scope items.
- **Mutually exclusive:** there are no duplicate subgoals or significant overlap.
- **Collectively exhaustive:** the combined subgoals completely define the overall goal.
- **One-to-one application mapping:** each subgoal has exactly one corresponding *App Used* entry identifying the primary application for that subgoal.
- **Matching order and count:** the two lists have exactly the same length and order.

Label	Definition
Yes	The subgoal list is coherent, goal-aligned and free of hallucinated items. Every subgoal has the correct one-to-one *App Used* entry.
No	$>= 1$ subgoal or *App Used* entry requires correction, addition, removal, or reordering.

Required action

If the label is No, update every affected subgoal or App Used. If the label is Yes, no action is needed.

D.5 Sequential Dependency (Boolean)

Evaluate whether subgoals are arranged in the logical execution order required to accomplish the overall goal. The list should describe a coherent task sequence, not an unordered collection where later subgoals do not depend on the completion of prior subgoals.

Label	Definition
Yes	Subgoals are in the correct sequence, and each naturally follows the preceding subgoal.
No	One or more subgoals are out of sequence, do not follow the required execution order, or do not depend on completion of prior subgoals.

Required action

If the label is No, reorder or edit the subgoal list to reflect the correct execution sequence where possible otherwise skip the task. If the label is Yes, no action is needed.

D.6 Stepwise State Labels

Apply the following four labels in order. Later labels depend on the earlier labels.

D.6.1 Subgoal Targeted (Multi-selection or None)

Select subgoals that the model is working toward in the current step. A step may be associated with one or more targeted subgoal. If it is unrelated to every defined subgoal, select None. For each step, only include subgoal(s) that either overlap with (e.g. subgoal 1 after the last step ends with subgoal 1) or directly follows the last subgoal from the prior step (e.g. subgoal 2 after the last step ends with subgoal 1).

D.6.2 Subgoal Feasibility (Boolean)

For the selected target, decide whether successful completion is possible in the current environment if the model continues taking appropriate actions.

Label	Definition
Yes	The environment, application, required resources, and system state permit the selected subgoal to be completed.
No	Completion is blocked by an application bug, sandbox limitation, unavailable required file/resource, missing precondition, UI/system failure, or another environment/system/application/resource limitation.

Feasibility rules

- If Targeted Subgoal is None, Feasibility defaults to No.
- If Feasibility is No, Progression and Completion must also be No for that step.
- Creating a substitute resource does not make the original subgoal feasible unless the substitute is a valid replacement for the required resource, such as retrieving the same original file from an appropriate source.

D.6.3 Subgoal Progression (Boolean)

Decide whether the current step produced observable, meaningful advancement toward the selected target. Progression reflects a state change, restoration, or completion - not merely an attempted action.

Label	Definition
Yes	The step observably advances the selected targeted subgoal.
No	The step makes no observable progress, is redundant, is blocked by infeasibility, or is unrelated to the selected target.

Progression rules

- If the targeted subgoal is partially or fully completed in the step, Progression is Yes.
- Verification generally does not count. It may count only when it materially advances evaluation of the subgoal, such as revealing that the previous approach was wrong or identifying an issue that changes the next action.
- Redundant verification of an already completed or already confirmed subgoal is No.
- If Feasibility is No, Progression is No.
- If Targeted Subgoal is None, Progression defaults to No.

D.6.4 SUBGOAL COMPLETION (BOOLEAN)

Mark Completion only on the step where the selected targeted subgoal is first fully achieved in the final evaluated attempt. A step may progress without completing the targeted subgoal.

Label	Definition
Yes	The selected targeted subgoal is first fully completed in this step.
No	The step does not fully complete the target, or the target was already completed earlier.

Completion rules

- Partial advancement remains Completion = No, even when Progression = Yes.
- Verification generally does not count. It receives Completion = Yes only if the subgoal objective is first fully achieved in that verification step.
- Post-completion cleanup (closing menus, dialogs, tabs, or windows) continues to target the relevant completed subgoal but receives Completion = No.
- If Feasibility is No, Completion is No.
- If Targeted Subgoal is None, Completion defaults to No.
- After a restart or method switch, assign Completion = Yes only at the first full completion in the final evaluated attempt.

D.6.5 DEPENDENCY QUICK REFERENCE

Condition	Required labels	Reason
Targeted Subgoal = None	Feasible = No Progress = No Completion = No	No defined milestone is being pursued.
Feasible = No	Progress = No Completion = No	Environmental or resource limitations prevent successful advancement/completion.
Partial state advance	Progress = Yes Completion = No	The target advances but is not fully satisfied.
First full achievement	Progress = Yes Completion = Yes	The target both advances and becomes complete.
Redundant verification	Progress = No Completion = No	No new state or material evaluation is produced.
Cleanup after completion	Target remains selected Completion = No	Cleanup is associated with the completed subgoal but does not complete it again.

D.7 REVIEWER WORKFLOW

Reviewer follows two stages: an independent blind evaluation and a comparison/review stage.

1. Perform the Blind annotation (for Reviewers). Use the Annotator rubric definitions and notes for all blind step-level scoring.

2. Submit the blind pass. Once the Annotator's work becomes visible, perform the task-level and step-level Agree/Disagree review.
3. If the step-level disagreement rate is greater than 5%, send the task back for rework.

Parameter	Agree (otherwise Disagree)
Subgoal Targeted	The subgoal(s) targetted are correct, or None is correctly selected for an unrelated step.
Subgoal Feasibility	The label correctly reflects if the environment permits successful completion.
Subgoal Progression	The label correctly reflects meaningful, goal-directed, observable state change.
Subgoal Completion	The label correctly marks the step where the target becomes fully complete.

Stepwise Reviewer threshold and comments

For each disagreed step, add a single consolidated comment summarizing every disagreed parameter in that step. If the overall step-level disagreement rate is greater than 5%, send the task for rework.

D.8 Shared Quality Assurance Strategy and Role Boundaries

- The Annotator produces the primary ratings and, where necessary, rationales/comments.
- Reviewer performs their blind pass without visibility into the Annotator's annotations, then performs the visible comparison review.
- Reviewer disagreement can trigger rework.

D.9 Worked Example

Suppose the goal is:

> Open `Presentation.PPT`, create a slide at the end, change its title to "Closing remarks," and save the result as `Presentation V2.PPT`.

One valid decomposition is:

Subgoal	Milestone	App Used
1	Open `Presentation.PPT`.	LibreOffice Impress
2	Create a slide at the end of the presentation.	LibreOffice Impress
3	Change the new slide title to "Closing remarks."	LibreOffice Impress
4	Save the file as `Presentation V2.PPT`.	LibreOffice Impress

For the step "Click the New Slide button":

Field	Example label
Subgoal Targeted	Subgoal 2, "Create a slide at the end of the presentation."
Subgoal Feasibility	Yes, if the presentation app and required file/state allow creation of a slide.
Subgoal Progression	Yes, if the click creates the slide or otherwise meaningfully advances slide creation.
Subgoal Completion	Yes only if this step first fully creates the required slide; otherwise No.

E Prompt Templates

Subgoal Decomposition `Decompose the overall goal into subgoals. Each subgoal will be individually used to assess task completion and should be used to assess subgoal completion in a binary fashion (i.e., cannot be partially fulfilled). Each subgoal should only contain one objective. If it has multiple objectives (such as when it needs do A and B), split it into multiple subgoals. Each subgoal should also only require one app to complete it - if it requires more than one app, split it into separate subgoals. Goal: {goal} Relevant Apps: {app_combo} Return a JSON array of subgoal objects. Each subgoal should: - Describe an independent component of the overall goal. - Be assessable in a binary fashion (completed or not completed). - Specify the app required to complete it. Return only [{"subgoal": "<subgoal_1>", "app":`

"<app_1>"}, {"subgoal": "<subgoal_2>", "app": "<app_2>"}, ...]
and nothing else.

F LLM JUDGE

Discussion on LLM Judge Design We started with a brute force approach to probe whether every subgoal is targeted by every step. If targeted, we can then iteratively probe whether the goal is feasible, progressed upon and completed. Assuming k steps (hundreds) and m subgoals (tens), we have a max of $O(k * m)$ API calls, which can all be done in parallel.

Not only is this approach demanding in API calls, it also does not apply the constraints of the sequentially-dependent subgoals found in OSWorld-Pro, where an agent will only target subgoals that follow the subgoals targeted in the prior step. Incorporating this memory feature (informing the LLM judge which subgoals were targeted by prior steps) means that it only requires $O(k)$ steps, but they need to be done in sequential order with much higher latency.

With this method, we realized that the LLM judge was basically re-using information from prior API calls. We found an approach to incorporate the information of all steps into a single API call ($O(1)$) containing up to hundreds of screenshots. While this means that only a limited number of models (GPT-5.6 family) can be used as LLM-Judges, we note that this is a common limitation in LLM-Judge based evaluations such as Arena Hard (LMSys, 2024) and GDPVal-AA (Artificial-Analysis, 2025). Specifically, these evaluations can only use some of the strongest models at time of release (e.g. Gemini-3-Pro or GPT-5 family). As the serving infrastructure of other models improve, this method can be applied directly on other models.

Below we show some snapshots of various LLM-Judge prompt templates that we used in our experiments including the final LLM-Judge prompt template. These templates represent some of our representative approach shifts with tens of variations in between them representing minor tweaks.

Initial Brute Force LLM-Judge Prompt Template Screenshot: <screenshot> Reasoning: <reasoning> Action: <action> Do the following attached screenshot, reasoning, and action suggest that the subgoal '<subgoal>' is <verb>? Only answer Yes or No where verb is one of ["targeted", "feasible", "being progressed towards", "completed"]

Later LLM-Judge Prompt Template with memory Screenshot: <screenshot> Reasoning: <reasoning> Action: <action> Which of the following subgoals are targeted by the attached screenshot, reasoning, and action? Subgoals: 0. <subgoal0> 1. <subgoal1> ... n. <subgoaln> Return only a JSON list with the index of the subgoal(s), including at least one subgoal where possible. Steps prior to this have targeted the following subgoal(s): 0. <subgoals-predicted-for-step0> 1. <subgoals-predicted-for-step1> ... k. <subgoals-predicted-for-stepk> Only include subgoal(s) that either overlap with (e.g. subgoal 1 after the last step ends with subgoal 1) or directly follows the last subgoal from the prior step (e.g. subgoal 2 after the last step ends with subgoal 1). Including subgoals that precede the subgoals targeted prior to the final subgoal in the prior step is NOT allowed (e.g. subgoal 0 when the last step targeted subgoal 1 OR subgoal 0 when the last step targeted subgoals [0, 1].

Final LLM Judge Prompt Template Step 0. Reasoning: <reasoning0> Action: <action0> <screenshot0> Step 1. Reasoning: <reasoning1> Action: <action1> <screenshot1> ... Step k. Reasoning: <reasoningk> Action: <actionk> <screenshotk> Which of the following subgoals are targeted by the attached screenshots (one for each step in sequential order), reasoning, and action? Subgoals: 0. <subgoal0> 1. <subgoal1> ... n. <subgoaln>

```
For each step, only include subgoal(s) that either overlap with
(e.g. subgoal 1 after the last step ends with subgoal 1) or
directly follows the last subgoal from the prior step (e.g.
subgoal 2 after the last step ends with subgoal 1). For each
step, including subgoals that precede the subgoals targeted
prior to the final subgoal in the prior step is NOT allowed
(e.g. subgoal 0 when the last step targeted subgoal 1 OR subgoal
0 when the last step targeted subgoals [0, 1]). In addition
to the above, indicate whether the subgoal is feasible, being
progressed towards, and completed for each step (Yes or No only).
Return only a nested JSON response with {len(actions)} items
with each item containing the index of the subgoal(s) for each
step under targeted, including at least one subgoal for each
step where possible. Take note to use items with more than one
subgoals per step sparingly as they are rare in practice. e.g {
0: { "targeted": [0], "feasible": "Yes", "progressed": "Yes",
"completed": "Yes" }, 1: { "targeted": [1], "feasible": "Yes",
"progressed": "Yes", "completed": "No" }, 2: { "targeted": [1,
2], "feasible": "Yes", "progressed": "Yes", "completed": "No" }
] }
```

G Inference Setup

LLM-Judge Cost Following ProfBench (Wang et al., 2026), we estimate the cost of running the full LLM-Judge evaluation, using the number of input and output tokens multiplied by their public API cost without caching (OpenAI, 2026a). We estimate fees based on the regular service tier at the lowest context length bucket. Early experiments also suggests that mean Macro-F1 is highly consistent, differing no more than 0.6% across three independent runs - therefore we only run with each judge once to save cost.

Benchmarking Details Following OSWorld (Xie et al., 2024), we perform one run for each model. We believe OSWorld opted for a single run due to the high cost of each run (up to thousands of US dollars per model) and minimal expected inter-run variance based on the high number of independent tasks (> 300). Across all models, we use the relevant model harnesses from OSWorld (XLANG-Lab, 2024), which defines the approach for inference details including sampling strategy (e.g. temperature and top-p), context management and system prompts. We set max turns at 250 (vs. 100 in OSWorld) due to longer horizon nature of our tasks. All models are evaluated at the highest reasoning effort possible.

Benchmarking Cost Prompts are cached in multi-step long-horizon tasks as they are optimal from cost considerations. In estimating cost, we use prices from OpenRouter (2025) including caching related fees and discounts as they contribute substantially to the eventual cost. For Anthropic models, we use the default 5-minute caching price. Across all models, we estimate fees based on the regular service tier at the lowest context length bucket.