HANDSOFFBENCH: CAN AGENTS AUTONOMOUSLY COMPLETE EVERYDAY PROJECTS FROM START TO FINISH?

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models are increasingly deployed as general-purpose assistants to fulfill everyday user requests across applications and services. As their performance on individual tasks improves, a central question is whether agents can reliably complete entire everyday projects without further human intervention. We introduce HandsOffBench, a benchmark comprising 32 everyday projects in a dynamic simulated world featuring 70 web applications, 30 stateful services, and code execution environments. Collected from individuals with diverse backgrounds, these projects require coordinating multiple subtasks across applications and services. Each project is adapted into an executable task with carefully designed subtask dependencies grounded in its practical requirements. These dependencies link later stages to earlier decisions, computed values, and generated artifacts, creating extended workflows whose execution can span thousands of steps. State checkers and rubric-based verifiers jointly assess subtask outcomes from the final environment state and generated artifacts, using an average of 195 scoring criteria per project. We evaluate a diverse set of frontier models covering different families, using both a minimal scaffold and widely used agent harnesses. The strongest evaluated model completes only 46.23% of projects with the minimal scaffold and 50.46% with harness support. To isolate the effect of execution horizon, we evaluate a subset of subtasks under identical scoring criteria and find substantially worse performance within complete projects than in isolation. These findings highlight the gap between subtask competence and reliable end-to-end completion of everyday projects, positioning HandsOffBench as a testbed for advancing model capabilities and agent harness design.

1 INTRODUCTION

Large language models (LLMs) are increasingly deployed as general-purpose assistants that interact with digital environments to fulfill everyday user requests (Wang et al., 2024; Yehudai et al., 2026; Xi et al., 2025). Many of these requests concern everyday projects that involve gathering information and producing deliverables across multiple applications and services. As models demonstrate continued progress in tool use, software interaction, and code execution (OpenAI, 2026; Anthropic, 2026a; Google DeepMind, 2026; Z.AI, 2026; Qwen Team, 2026), a central question is whether they can sustain this competence throughout an entire project without further human intervention.

Despite progress in agent evaluation, the reliability of agents in executing long-horizon everyday projects end-to-end warrants further investigation. Existing benchmarks assess agent capabilities in software engineering (Jimenez et al., 2024; Deng et al., 2025), information seeking (Wei et al., 2025; Yoran et al., 2024), computer interaction (Gou et al., 2025; Xie et al., 2024), and tool-mediated workflows (Yao et al., 2024; Barres et al., 2025; He et al., 2026). More recent work extends evaluation to specialized projects in research replication (Starace et al., 2025), machine-learning engineering (Chan et al., 2025), workplace activities (Xu et al., 2024), and economically valuable industry workflows (Sun et al., 2026). In these settings, extended execution can involve demanding domain-specific reasoning and navigation of specialized environments. Everyday projects provide a complementary setting in which completing the overall project involves coordinating many routine

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.

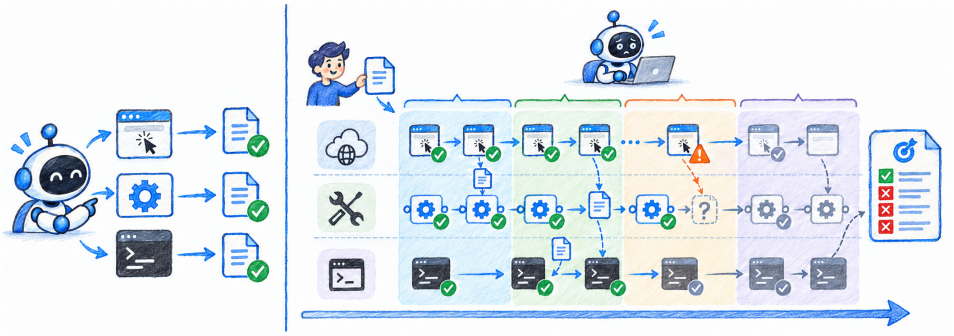


Figure 1: Agents can perform routine operations in isolation (left), yet struggle to coordinate them across long-horizon everyday projects without human intervention (right).

operations across stages, applications, and services, as illustrated in Figure 1. Evidence of declining performance over longer horizons (Lu et al., 2026; Motwani et al., 2026) further motivates examining whether abilities demonstrated on individual operations persist throughout a complete project. Systematically assessing this question requires complex tasks and diverse environments.

To evaluate how reliably agents complete everyday projects without human intervention, we collect projects from individuals with diverse backgrounds and adapt them into executable tasks. Each project represents a complete user objective involving commonly used applications and tools and is decomposed into subtasks with explicit requirements and deliverables. Grounded in practical requirements, we carefully design dependencies among subtasks so that later stages build on earlier decisions, computed values, and generated artifacts. These dependencies extend the execution horizon by requiring agents to preserve and combine intermediate outcomes across successive stages.

Building on this design, we introduce HandsOffBench, a benchmark comprising 32 everyday projects across diverse domains and a total of 388 subtasks, summarized in Figure 2. The simulated world in HandsOffBench includes 70 web applications, 30 stateful services, and general-purpose code execution environments. End-to-end runs on these projects can span thousands of interaction steps and consume millions of tokens. To approximate real-world conditions, application and service states can change during execution in a subset of tasks, as new information becomes available or existing records are updated. These updates can invalidate earlier assumptions or intermediate results, requiring agents to reassess earlier conclusions and revise affected results. Each project is evaluated once the agent declares the project complete and stops, measuring how fully the agent has satisfied the user objective. We combine state checkers with rubric-based verifiers to assess the final environment state and generated artifacts, producing fine-grained scores for individual subtasks.

We evaluate a broad set of frontier models on HandsOffBench using a shared minimal scaffold, HandsOff-Harness, and compare selected backbones across widely used agent harnesses. The strongest model completes only 46.23% of projects with the minimal scaffold; harness support improves performance, but even the best model–harness configuration completes only 50.46%. To understand this limited reliability, we examine subtask outcomes across workflow stages and find that scores generally decline toward later stages. A matched comparison further shows that the same subtasks receive higher scores in isolation than within complete projects, indicating challenges beyond individual subtask difficulty. Analysis of execution traces identifies recurring failures in task control, state tracking, and cross-stage consistency. These findings reveal a gap between competence on individual operations and reliable completion of everyday projects, positioning HandsOffBench as a testbed for improving autonomous execution through advances in models and harnesses.

2 Related Work

2.1 Benchmarking LLM Agent

Systematic evaluation provides a common basis for measuring progress in LLM agents and identifying limitations in task execution. Existing agent benchmarks cover tool-mediated service work-

flows (Yao et al., 2024; Barres et al., 2025), information seeking (Wei et al., 2025; Yoran et al., 2024), web interaction (Zhou et al., 2024; Yao et al., 2022; Gou et al., 2025), computer use (Xie et al., 2024), software engineering (Jimenez et al., 2024; Deng et al., 2025), and assessing general-assistant capabilities across heterogeneous tasks and environments (Mialon et al., 2024; Liu et al., 2024). Recent benchmarks also extend evaluation to specialized projects in research replication (Starace et al., 2025), machine-learning engineering (Chan et al., 2025), workplace activities (Xu et al., 2024; Boisvert et al., 2024), and economically valuable industry workflows (Sun et al., 2026) The deployment of agents as personal assistants also motivates evaluating everyday user requests that require coordinated interaction across applications and services. Related benchmarks examine these tasks through interactive service scenarios (He et al., 2026), code-based interaction across applications (Trivedi et al., 2024), long-horizon computer-use workflows (Yuan et al., 2026), and proactive assistance in evolving life mock service (Xiaohongshu Dots Studio & Evolvent AI, 2026). HandsOffBench complements these efforts by focusing on whether agents reliably complete everyday projects in a dynamic simulated world without further human intervention

2.2 Long-Horizon Execution in LLM Agents

Sustaining reliable execution over long sequences of interactions is a central challenge for LLM agents, motivating research in both harness design and model training. On the harness side, tool and execution environment interfaces provide access to external systems (Wang et al., 2025a; Yang et al., 2024), while interleaved reasoning and action (Yao et al., 2023b), search (Yao et al., 2023a), and feedback-based reflection (Shinn et al., 2023) guide multi-step decisions. For longer runs, context management and workflow orchestration help preserve relevant information and coordinate successive stages of execution (Anthropic, 2025; OpenAI, 2025; Wu et al., 2023; Young, 2025). On the training side, models learn to sustain reasoning and tool use over long horizons through supervised fine-tuning on interaction trajectories (Zeng et al., 2024; Chen et al., 2023; Li et al., 2026b; Liu et al., 2025). Reinforcement learning further improve model's reasoning, search, tool use, and multi-turn decision-making using feedback from task outcomes (DeepSeek-AI, 2025; Jin et al., 2025; Feng et al., 2025; Wang et al., 2025b). To support training on longer trajectories, recent work also addresses optimization stability and context constraints through advances in training infrastructure and learning algorithms (Hou et al., 2026; Li et al., 2026a). Despite these advances, empirical studies report that performance can decline as task horizons increase, even in models that support large context windows (Zhou et al., 2025; Lu et al., 2026; Motwani et al., 2026). These findings motivate our evaluation of whether competence on individual subtasks translates into reliable autonomous completion of everyday projects across applications and services in dynamic simulated environments.

3 HandsOffBench

In this section, we present HandsOffBench, a benchmark for evaluating how reliably agents complete everyday projects from start to finish without human intervention. Figure 2 illustrates the benchmark design, and Table 1 summarizes its statistics by project domain. We describe the task formulation, simulated world, construction procedure, and scoring protocol below.

3.1 Task Formulation

Each project in HandsOffBench requires an agent to fulfill an everyday user objective through long-horizon execution in a simulated digital world. We represent a project as

$$\mathcal{T}_i = (q_i, s_i^0, \mathcal{C}_i), \tag{1}$$

where q_i specifies the overall user request, s_i^0 is the initial environment state, and $\mathcal{C}_i$ contains the project-specific evaluation criteria.

Each project comprises subtasks corresponding to meaningful stages of execution, with dependencies linking later stages to earlier outcomes. These subtasks and their dependencies form a directed acyclic graph:

$$V_i = v_{i1}, \ldots, v_{im_i}, \qquad G_i = (V_i, E_i), \tag{2}$$

where an edge $(v_{ij}, v_{ik}) \in E_i$ indicates that v_{ik} depends on outcomes produced by v_{ij}. Each subtask specifies requirements and deliverables that may involve multiple routine operations across applications and services, such as sending an email or recording information in a spreadsheet. Its outcomes,

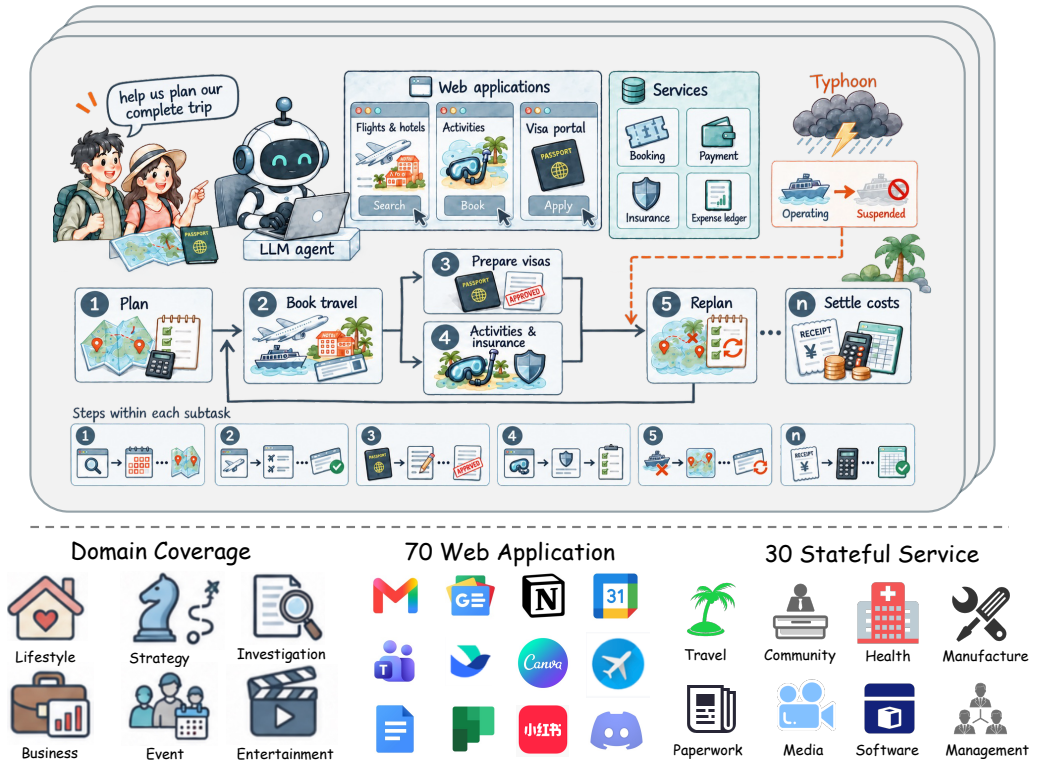


Figure 2: Overview of HandsOffBench. The top shows an example project of organizing a trip end-to-end, from itinerary planning and bookings to adapting to unexpected disruptions and settling costs. Bottom shows the coverage of everyday project domains and the simulated world, featuring 70 web applications and 30 stateful services.

including environment changes and generated artifacts, are assessed separately using the associated evaluation criteria $\mathcal{C}_{ij} \subseteq \mathcal{C}_i$. An edge $(v_{ij}, v_{ik}) \in E_i$ indicates that v_{ik} uses outcomes produced by v_{ij}. Branching and merging dependencies capture how intermediate outcomes are reused and combined across the project. Revisions to earlier outcomes may also call for corresponding updates to subsequent results.

Agents pursue the complete user objective without further human intervention. Let π denote the execution policy induced by the model and its harness:

$$a_t \sim \pi(\cdot \mid q_i, \mathcal{D}_i, h_t), \qquad a_t \in \mathcal{A}_i \cup \{\texttt{STOP}\}, \tag{3}$$

where $\mathcal{D}_i$ provides application and tool descriptions, $\mathcal{A}_i$ is the available action set, and $h_t = (o_0, a_0, \ldots, a_{t-1}, o_t)$ records previous actions and observations. The agent system determines the execution order and chooses when to terminate by selecting `STOP`. The resulting outcomes are then evaluated to determine how fully the project has been completed.

3.2 Simulated World

Everyday projects often span multiple applications and services. HandsOffBench provides a simulated digital world in which agents pursue these projects through web interaction, structured tool use, and general-purpose code execution.

Web Applications. Web applications support project activities such as information seeking and content creation. We select our application pool based on occupational taxonomies and empirical patterns of AI use across economic tasks (Gregory et al., 2019; Handa et al., 2025). Building on Wang et al. (2026), we extend its mock applications to better approximate the functionality of their real-world counterparts and support the operations and state changes involved in our projects. The

Table 1: Domain coverage and project statistics of HandsOffBench.

Domain	Projects	Per-project averages				Dynamic events	
		Subtasks	Apps	Services	Criteria	Projects	Events
Lifestyle	6	11.7	7.0	4.0	170	2	7.5
Entertainment	6	13.0	11.3	7.3	235	5	5.6
Events	6	10.7	9.7	5.3	148	4	7.0
Business	7	13.4	10.4	7.6	231	2	15.0
Strategy	5	12.0	7.2	5.4	196	3	7.7
Investigation	2	11.0	11.5	7.5	158	1	2.0
Overall	32	12.1	9.4	6.1	195	17	7.4

resulting pool comprises 70 locally hosted web applications spanning diverse scenarios. Agents observe application interfaces through screenshots and interact using mouse and keyboard actions.

Services and Tool Interfaces. Projects also involve retrieving information and carrying out transactions through APIs provided by different platforms. Following prior benchmarks (Barres et al., 2025; Zheng et al., 2026), we provide stateful services that agents access through structured tool calls. The service backends maintain task-specific data tables, while their tool interfaces define the agent's action space for interacting with these services. Our environment includes 194 services spanning diverse everyday scenarios, exposing a diverse set of tool interfaces for information retrieval, record modification, and service-specific transactions.

Dynamic World. Real-world information and service states can evolve during project execution, affecting both the validity of earlier decisions and the conditions for subsequent stages. To approximate these dynamics, a subset of tasks includes events that introduce new information or supersede existing records as agents make progress. These changes may require agents to reassess prior conclusions, adjust upcoming actions, and revise affected records or artifacts. Because subtasks build on earlier outcomes, adapting to a change may also involve identifying and updating dependent results across multiple stages. This setting evaluates whether agents can track evolving conditions and keep their decisions and deliverables consistent while pursuing the complete user objective.

3.3 Task Construction

We adapt everyday projects into executable tasks through manual annotation. The annotation team specifies requirements, prepares environment content, and designs dependencies among subtasks.

Domain Selection. We collect projects from individuals with diverse backgrounds, covering lifestyle, entertainment, events, business, strategy, and investigation. These projects represent everyday objectives that users may delegate to general-purpose assistants, with concrete requirements and deliverables supported by our simulated applications and services.

Environment Construction. We populate each project's environment with documents, application records, and service data. Relevant materials appear alongside realistic distractors, including outdated documents, similarly named entities, and records outside the requested scope. Information is distributed across sources and paginated collections, requiring agents to identify applicable evidence, retrieve all relevant records, and combine their findings through computation and reasoning.

Project Dependencies. The annotation team designs dependencies grounded in practical project requirements, linking later stages to earlier decisions, computed values, and artifacts. Sequential chains, branching, and merging require agents to preserve, reuse, and combine intermediate outcomes while maintaining consistency throughout execution.

The resulting benchmark comprises 32 everyday projects and 388 individually verifiable subtasks, with 9–17 subtasks per project (12.1 on average).

3.4 Evaluation Protocol

We evaluate project completion through subtask outcomes retained in the final environment state and generated artifacts. State checkers verify these outcomes against task-specific criteria, covering application and service changes as well as facts and computed values in artifacts. An LLM-based verifier reviews the checker assessments to help identify rule-based misjudgments. Each subtask contains an average of 15 scoring criteria, amounting to approximately 195 criteria per project.

For subtask v_{ij}, let $\widehat{\mathcal{C}}_{ij} \subseteq \mathcal{C}_{ij}$ denote its scored criteria and $n_{ij} = |\widehat{\mathcal{C}}_{ij}|$. The subtask score is the fraction of satisfied criteria, while the project score averages criterion satisfaction across all scored subtasks:

$$\begin{aligned} R_{ij} &= \frac{1}{n_{ij}} \sum_{c \in \widehat{\mathcal{C}}_{ij}} c(x_i^{\text{final}}), \qquad n_{ij} > 0, \\ R_i &= \frac{\sum_{j:n_{ij}>0} n_{ij} R_{ij}}{\sum_{j:n_{ij}>0} n_{ij}}, \end{aligned} \tag{4}$$

where x_i^{final} denotes the final environment state and generated artifacts, and $c(x_i^{\text{final}}) \in \{0, 1\}$ indicates whether criterion c is satisfied.

4 Experiments

4.1 Experimental Setup

Models and Harnesses. We evaluate a diverse set of state-of-the-art models spanning different release generations and model scales, covering the GPT (OpenAI, 2026), Claude (Anthropic, 2026b), Gemini (Google DeepMind, 2026), Kimi (Moonshot AI, 2026), DeepSeek (DeepSeek-AI, 2026), and GLM (GLM-5 Team, 2026) families. For a controlled model comparison, all models run within HandsOff-Harness, our minimal scaffold that executes model-issued actions and returns observations from the simulated world. We additionally evaluate two widely used agent harnesses, Codex (OpenAI, 2025) and Claude Code (Anthropic, 2025). Harness comparisons use shared backbones, with task instances, environment interfaces, and execution budgets held fixed.

Evaluation Metrics. We report fine-grained project performance and two measures of resource use. Following the scoring rules in Section 4.3.3, Score (%) is the mean project score R_i, expressed as a percentage, with each project weighted equally. Steps counts executed environment actions. Token (M) measures input and output tokens, in thousands, summed over all model calls within a run, including harness-internal calls. Both resource metrics are averaged over all evaluated projects.

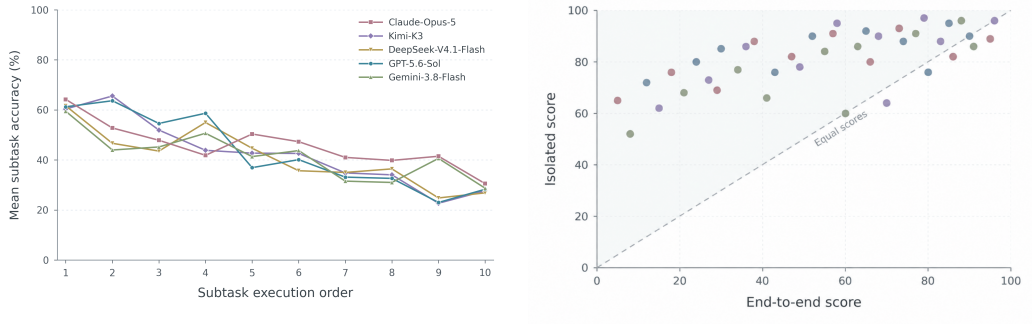
Implementation Details. Each run is limited to 72 hours of wall-clock time, 128K execution steps and 128K output tokens, with no overall token limit. The agent starts from the task's initial environment state, receives the complete user request and application and tool descriptions, and operates without human intervention until it terminates or reaches either limit. State checkers evaluate the final environment state and generated artifacts, including those retained when an execution limit is reached. We use Claude-Opus-5 as an LLM verifier to double-check these assessments for potential rule-based misjudgments.

4.2 Main Results

Agents struggle to complete everyday projects autonomously. No evaluated agent fully completes a single long-horizon everyday project without human intervention. As shown in Table 2, under the HandsOff scaffold, Claude-Opus-5.5 achieves the highest fine-grained Score of 48.17%, while Kimi-K3 leads the evaluated open-source models at 42.00%. These scores indicate partial fulfillment of project requirements, but do not translate into successful end-to-end completion. Although individual operations are often routine—such as reading an email or adding a row to a spreadsheet—coordinating them into a complete workflow remains challenging. This difficulty persists despite substantial execution effort: Claude-Opus-5.5 averages 1,129.5 steps and 174.6M tokens per project. The results highlight the gap between performing individual operations and sustaining progress until the full user objective is satisfied.

Scaffold	Model	Open-Source	Score (%)	Steps	Tokens (M)
HandsOff	GPT-6-Sol	✗	46.23	1,174.8	178.4
	GPT-5.6-Sol	✗	42.70	1,236.5	185.7
	GPT-5.6-Luna	✗	34.86	914.3	112.6
	GPT-5.6-Terra	✗	38.41	1,068.7	147.9
	Claude-Opus-5	✗	44.80	1,086.6	162.0
	Kimi-K3	✓	42.00	1,193.4	171.3
	DeepSeek-V4.1-Flash	✓	41.40	1,348.2	154.8
	Claude-Opus-5.5	✗	48.17	1,129.5	174.6
	Gemini-3.8-Flash	✗	40.90	1,017.9	128.5
Codex	GPT-6-Sol	✗	49.38	1,326.4	206.8
	GPT-5.6-Sol	✗	45.26	1,408.7	219.3
	GPT-5.6-Luna	✗	33.94	1,052.6	138.7
	GPT-5.6-Terra	✗	40.07	1,214.3	169.4
Claude Code	Claude-Opus-5	✗	48.63	1,249.8	191.5
	Kimi-K3	✓	43.72	1,367.1	208.6
	Claude-Opus-5.5	✗	50.46	1,193.6	184.2
	Gemini-3.8-Flash	✗	40.21	1,186.5	153.7

Table 2: Main results on HandsOffBench. Score measures fine-grained project performance; Steps and Token report average execution costs.

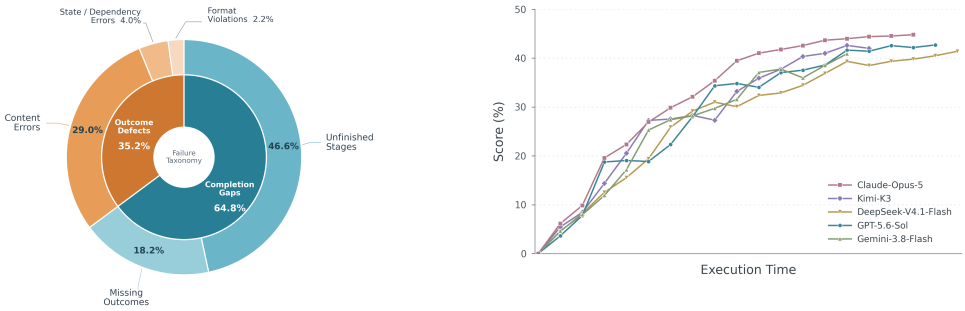


(a) Subtask accuracy by execution order.

(b) Isolated versus end-to-end execution.

Figure 3: Analysis of long-horizon execution on HandsOffBench. (a) Mean subtask accuracy by execution order. (b) Paired subtask scores under isolated and end-to-end execution for a fixed model. Colors distinguish source projects, and points above the dashed diagonal indicate higher scores in isolation.

Harness support yields uneven gains and does not ensure reliable completion. We next examine whether agent harnesses help close this gap. Holding the backbone and task instances fixed, Codex improves GPT-6-Sol's Score from 46.23% to 49.38%, while Claude Code improves Claude-Opus-5.5's Score from 48.17% to 50.46%. These gains show that execution support can improve fine-grained project performance without changing the underlying model. However, the benefits are not consistent across models: GPT-5.6-Luna declines from 34.86% to 33.94% with Codex, and Gemini-3.8-Flash declines from 40.90% to 40.21% with Claude Code. Moreover, every evaluated harness configuration consumes more steps and tokens than its HandsOff counterpart. Even the strongest configuration, Claude-Opus-5.5 with Claude Code, achieves only 50.46% Score and fails to fully complete any project. These results reveal substantial room for improving both the reliability and efficiency of existing harnesses on long-horizon everyday workflows.



(a) Failure taxonomy.

(b) Score over execution time.

Figure 4: Failure modes and execution progress on HandsOffBench. (a) Distribution of classified failures across completion gaps and outcome defects, with a finer-grained breakdown into five subcategories. (b) Project scores at successive execution checkpoints.

4.3 Analysis

4.3.1 Effect of Horizon

Our main results show that even the strongest evaluated agents struggle to complete long-horizon everyday projects composed of routine operations. In this section, we examine how this challenge relates to execution horizon by analyzing performance across workflow stages and comparing the same subtasks in isolated execution and end-to-end execution.

Performance across Workflow Stages. We first examine whether agents sustain their accuracy across successive subtasks within a project. For each execution-order position k, we average the individual accuracy of the k-th executed subtask across projects with an available score at that position, considering the first ten positions for coverage. Figure 3a shows an overall decline in accuracy as execution progresses. For example, Claude-Opus-5's mean subtask accuracy falls from 64.22% at the first position to 30.54% at the tenth, a decrease of 33.68 percentage points despite occasional recoveries at intermediate positions. Subtask order follows workflow dependencies, and intrinsic difficulty is not explicitly tied to position in our task design. This pattern is consistent with difficulties in sustaining long-horizon execution: agents must retain earlier decisions, correctly reuse intermediate results, and coordinate dependent actions while tracking outstanding requirements.

Isolated versus End-to-End Execution. To examine the additional demands of project-level execution, we compare the same subtasks in isolation and within complete projects. Each isolated run starts from a gold-standard environment snapshot with prerequisite states and artifacts correctly established, and receives standalone instructions containing only the target subtask's requests. The model, scaffold, and scoring criteria are held fixed. In Figure 3b, most paired observations lie above the equal-score diagonal, indicating that the evaluated model achieves higher scores in isolation. Because the target subtasks and scoring criteria remain unchanged, this gap suggests that intrinsic subtask difficulty alone does not explain the observed failures. Within complete projects, agents must additionally establish valid prerequisites, retain relevant information, and preserve intermediate artifacts across dependent stages. The comparison thus exposes a gap between solving a subtask under prepared conditions and successfully carrying it out as part of an extended workflow.

4.3.2 Failure Modes

Completion gaps are the predominant failure mode in long-horizon everyday projects, accounting for 64.8% of the classified failure instances in Figure 4a. When carrying out extended workflows, agents often leave requirements unfulfilled or terminate execution before all required work is complete. A finer-grained breakdown shows that unfinished stages account for 46.6% of failures, indicating that substantial portions of a project remain incomplete. Missing outcomes contribute another 18.2%, where requested deliverables or state changes are absent. In observed runs, premature termination

can be agent-initiated rather than imposed by external budget cutoffs, highlighting the difficulty of sustaining execution until the full request is satisfied. The remaining 35.2% involve outcome defects. Content errors account for 29.0% of all failures, where produced results contain incorrect or incomplete substantive content. State and dependency errors contribute 4.0%, involving incorrect states or improperly handled dependencies between subtasks. Format violations account for the remaining 2.2%, where outputs fail to meet the required format. These failures show that producing an outcome does not necessarily satisfy the corresponding requirement: its content, relationship to other project states, and format must also be correct.

4.3.3 Progress over Execution Time

We further examine how execution time translates into progress toward the complete project objective. We track scores at successive execution checkpoints and average them over the all projects. Figure 4b illustrates how progress varies over execution: scores increase substantially early on, while later gains become smaller. For example, Claude-Opus-5 gains 29.85 % during an early execution phase, compared with only 3.02 % performance gain during a late phase spanning the same number of checkpoint intervals, reaching a final mean score of 44.80%. This slowdown may partly reflect error propagation and goal drift. Early errors can propagate through subtask dependencies, causing later stages to build on incorrect intermediate results and limiting the progress achieved by subsequent actions. Over prolonged execution, agents may also lose track of outstanding requirements and gradually deviate from the original project objective, directing effort toward actions that contribute little to completing the remaining work.

5 Limitations

Our evaluation covers multiple model families and representative agent harnesses, but the cost of long-horizon execution limits the breadth of configurations and repeated runs we can examine. The reported results should therefore be interpreted within the evaluated model versions, harness configurations, and execution budgets. Benchmark construction requires substantial effort to collect everyday projects, implement executable environments, and verify dependencies and outcomes. This limits the scale and diversity of the current task set. Although the projects span varied backgrounds and scenarios, they do not exhaust the range of everyday workflows or user needs. We are expanding the collection to improve coverage in future releases. Finally, our simulated world captures only a subset of the variability encountered in live applications and services, including changes in interfaces, service behavior, and user requirements. Generalization to real deployments therefore requires further investigation. While fine-grained criteria support systematic assessment of task outcomes, they may not capture every aspect of artifact quality, user satisfaction, or the consequences of intermediate actions. Benchmark performance should thus be interpreted as evidence of capability under the specified evaluation conditions, without implying readiness for unsupervised deployment.

6 Conclusion

We introduced HandsOffBench, a benchmark for evaluating whether LLM agents can complete everyday projects end-to-end without human intervention. We collected projects from individuals with diverse backgrounds and adapted them into executable tasks in a simulated digital world. These tasks connect interdependent stages across applications and services, requiring agents to preserve and reuse earlier decisions, computed values, and generated artifacts throughout execution. Fine-grained verification of subtask outcomes enables us to examine not only overall project completion but also how agent performance evolves across extended workflows. Evaluations with a minimal scaffold and agent harnesses show that reliable project completion remains challenging. Matched comparisons reveal lower performance on the same subtasks within complete projects than in isolation, while trace analysis identifies error propagation, inaccurate state tracking, and premature termination as recurring failures. Together, these results demonstrate that competence on individual operations does not yet reliably compose into autonomous completion of extended projects. These findings position HandsOffBench as a testbed for advancing models and agent harnesses toward reliable autonomous completion of everyday projects.

AI USE STATEMENT

We used generative AI tools to assist with manuscript revision, literature exploration, discussions of benchmark design, and the preparation of figures and plotting code. The evaluated language models and agent harnesses are experimental systems whose actions and outputs are assessed within the benchmark. Task requirements, environment content, and subtask dependencies are manually annotated by a dedicated data team, as described in Section 3.3. The scoring protocol uses deterministic state checks and rubric-based checks of generated artifacts, as specified in Section 3.4. The authors take responsibility for the benchmark design, AI-assisted materials, experimental analyses, and final manuscript.

ETHICS STATEMENT

HandsOffBench evaluates autonomous agents in a simulated digital world comprising locally hosted web applications, stateful services, and code execution environments. This setting supports the study of extended workflows and their failures within controlled task environments. Project descriptions and environment materials may raise privacy or confidentiality concerns, which should be addressed in their collection, adaptation, and release. More capable autonomous agents can also amplify the consequences of incorrect decisions or unauthorized actions over long workflows. We therefore emphasize permission boundaries, traceable actions, and opportunities for human intervention in downstream deployments. Performance on the benchmark does not establish that an agent can safely operate without oversight in real-world applications.

REPRODUCIBILITY STATEMENT

Section 3 describes the task formulation, simulated world, task-construction procedure, and scoring protocol. Section 4.1 defines the evaluation metrics and the comparison between models and harnesses. Section 4.3.1 describes the matched comparison between isolated subtask execution and end-to-end project execution, including the use of shared scoring criteria and correctly established prerequisite outcomes for isolated runs. These definitions specify how task outcomes are assessed and how the principal comparisons are constructed, providing a basis for reproducing the benchmark evaluation.

REFERENCES

Anthropic. Claude Code: An agentic coding tool. https://www.anthropic.com/product/claude-code, 2025.

Anthropic. Introducing Claude Opus 4.8. https://www.anthropic.com/news/claude-opus-4-8, 2026a.

Anthropic. Introducing Claude Opus 5. https://www.anthropic.com/news/claude-opus-5, 2026b.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025.

Léo Boisvert, Megh Thakkar, Maxime Gasse, Massimo Caccia, Thibault Le Sellier De Chezelles, Quentin Cappart, Nicolas Chapados, Alexandre Lacoste, and Alexandre Drouin. WorkArena++: Towards compositional planning and reasoning-based common knowledge work tasks. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0195. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/0b82662b6c32e887bb252a74d8cb2d5e-Abstract-Datasets_and_Benchmarks_Track.html.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, et al. MLE-bench: Evaluating machine learning agents on machine learning engineering. In *International Conference on Learning Representations (ICLR)*, 2025.

Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. FireAct: Toward language agent fine-tuning, 2023. URL https://arxiv.org/abs/2310.05915.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Xiang Deng, Jeff Da, Edwin Pan, Yannis Yiming He, Charles Ide, Kanak Garg, Niklas Lauffer, et al. SWE-Bench Pro: Can AI agents solve long-horizon software engineering tasks? *arXiv preprint arXiv:2509.16941*, 2025.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, et al. ReTool: Reinforcement learning for strategic tool use in LLMs. *arXiv preprint arXiv:2504.11536*, 2025.

GLM-5 Team. GLM-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Google DeepMind. Gemini 3.1 Pro model card. https://deepmind.google/models/model-cards/gemini-3-1-pro/, 2026.

Boyu Gou, Zanming Huang, Yuting Ning, Yu Gu, Michael Lin, Weijian Qi, Andrei Kopanev, Botao Yu, Bernal Jiménez Gutiérrez, Yiheng Shu, Chan Hee Song, Jiaman Wu, Shijie Chen, Hanane Nour Moussa, Tianshu Zhang, Jian Xie, Yifei Li, Tianci Xue, Zeyi Liao, Kai Zhang, Boyuan Zheng, Zhaowei Cai, Viktor Rozgic, Morteza Ziyadi, Huan Sun, and Yu Su. Mind2Web 2: Evaluating agentic search with agent-as-a-judge. *arXiv preprint arXiv:2506.21506*, 2025. URL https://arxiv.org/abs/2506.21506.

Christina Gregory, Phil Lewis, Pamela Frugoli, and Alexander Nallin. Updating the O*NET-SOC taxonomy: Incorporating the 2018 SOC structure. Technical report, National Center for O*NET Development and U.S. Department of Labor, August 2019. URL https://www.onetcenter.org/reports/Taxonomy2019.html.

Kunal Handa, Alex Tamkin, Miles McCain, Saffron Huang, Esin Durmus, Sarah Heck, Jared Mueller, Jerry Hong, Stuart Ritchie, Tim Belonax, Kevin K. Troy, Dario Amodei, Jared Kaplan, Jack Clark, and Deep Ganguli. Which economic tasks are performed with AI? evidence from millions of Claude conversations, 2025. URL https://arxiv.org/abs/2503.04761.

Wei He, Yueqing Sun, Hongyan Hao, Xueyuan Hao, Zhikang Xia, Qi Gu, Chengcheng Han, Dengchang Zhao, Hui Su, Kefeng Zhang, Man Gao, Xi Su, Xiaodong Cai, Xunliang Cai, Yu Yang, and Yunke Zhao. VitaBench: Benchmarking LLM agents with versatile interactive tasks in real-world applications. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=rtcX9q0Baz.

Zhenyu Hou, Yujiang Li, Jie Tang, and Yuxiao Dong. Single-rollout asynchronous optimization for agentic reinforcement learning, 2026. URL https://arxiv.org/abs/2607.07508.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, 2024.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Yujiang Li, Zhenyu Hou, Yi Jing, Jie Tang, and Yuxiao Dong. CompactionRL: Reinforcement learning with context compaction for long-horizon agents, 2026a. URL https://arxiv.org/abs/2607.05378.

Zhuofeng Li, Dongfu Jiang, Xueguang Ma, Haoxiang Zhang, Ping Nie, Yuyu Zhang, Kai Zou, Jianwen Xie, Yu Zhang, and Wenhu Chen. OpenResearcher: A fully open pipeline for long-horizon deep research trajectory synthesis, 2026b. URL https://arxiv.org/abs/2603.20278.

Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, Jiayuan Song, Zhengmao Zhu, Wenhu Chen, Pengyu Zhao, and Junxian He. WebExplorer: Explore and evolve for training long-horizon web agents, 2025. URL https://arxiv.org/abs/2509.06501.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations (ICLR)*, 2024.

Yi Lu, Jianing Wang, Linsen Guo, Wei He, Hongyin Tang, Tao Gui, Xuanjing Huang, Xuezhi Cao, Wei Wang, and Xunliang Cai. R-Horizon: How far can your large reasoning model really go in breadth and depth? In *International Conference on Learning Representations (ICLR)*, 2026.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: A benchmark for general AI assistants. In *International Conference on Learning Representations (ICLR)*, 2024.

Moonshot AI. Kimi K3 tech blog: Open frontier intelligence. https://www.kimi.ai/blog/kimi-k3, 2026.

Sumeet Ramesh Motwani, Daniel Nichols, Charles London, Peggy Li, Fabio Pizzati, et al. LongCoT: Benchmarking long-horizon chain-of-thought reasoning. *arXiv preprint arXiv:2604.14140*, 2026.

OpenAI. Introducing Codex. https://openai.com/index/introducing-codex/, 2025.

OpenAI. GPT-5.5 system card. https://openai.com/index/gpt-5-5-system-card/, 2026.

Qwen Team. Qwen3.7: The agent frontier. https://www.alibabacloud.com/blog/qwen3-7-the-agent-frontier_603154, 2026. Qwen3.7-Max: reasoning agent model with 1M-token context, API-only.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, et al. PaperBench: Evaluating AI's ability to replicate AI research. *arXiv preprint arXiv:2504.01848*, 2025.

Yiyou Sun, Xinyang Han, Weichen Zhang, Yuanbo Pang, Tianyu Wang, et al. Agents' last exam. *arXiv preprint arXiv:2606.05405*, 2026.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Bowen Wang, Dunjie Lu, Junli Wang, Tianyi Bai, Shixuan Liu, Zhipeng Zhang, Haiquan Wang, Hao Hu, Tianbao Xie, Shuai Bai, Dayiheng Liu, Que Shen, Junyang Lin, and Tao Yu. CUA-Gym: Scaling verifiable training environments and tasks for computer-use agents, 2026. URL https://arxiv.org/abs/2605.25624.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-Rong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6), 2024.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, et al. OpenHands: An open platform for AI software developers as generalist agents. In *International Conference on Learning Representations (ICLR)*, 2025a.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, et al. RAGEN: Understanding self-evolution in LLM agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025b.

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

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023. URL https://arxiv.org/abs/2308.08155.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, et al. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 2025. arXiv:2309.07864.

Xiaohongshu Dots Studio and Evolvent AI. VibeLifeBench: Can your life agent be proactive and persistent in a living world?, 2026. URL https://arxiv.org/abs/2608.10875.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, et al. OS-World: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Frank F. Xu, Yufan Song, Boxuan Li, Yuxuan Tang, et al. TheAgentCompany: Benchmarking LLM agents on consequential real world tasks. *arXiv preprint arXiv:2412.14161*, 2024.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023b.

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

Asaf Yehudai, Lilach Eden, Alan Li, Guy Uziel, Yilun Zhao, Roy Bar-Haim, Arman Cohan, and Michal Shmueli-Scheuer. A survey on evaluation of LLM-based agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 26690–26714. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1330. URL https://aclanthology.org/2026.findings-acl.1330/.

Ori Yoran, Samuel Joseph Amouyal, Chaitanya Malaviya, Ben Bogin, Ofir Press, and Jonathan Berant. AssistantBench: Can web agents solve realistic and time-consuming tasks? In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

Justin Young. Effective harnesses for long-running agents. Anthropic Engineering, November 2025. URL https://www.anthropic.com/engineering/effective-harnesses-for-long-running-agents.

Mengqi Yuan, Zilong Zhou, Xinzhuang Xiong, Weiming Wu, Jiayang Sun, Jiamin Song, Kaiqian Cui, Bowen Wang, Haoyuan Wu, Yitong Li, Dunjie Lu, Haikong Lu, Qi Zhen, Xinyuan Wang, Jiaqi Deng, Yuhao Yang, Cheng Chen, Boyuan Zheng, Alex Su, Xiao Yu, Hao Zou, Saaket Agashe, Xing Han Lu, Manpreet Kaur, Zhengyang Qi, Vincent Sunn Chen, Frederic Sala, Dayiheng Liu, Junyang Lin, Zhou Yu, Yu Su, Siva Reddy, Xin Eric Wang, Peng Qi, Tianbao Xie, and Tao Yu. OSWorld 2.0: Benchmarking computer use agents on long-horizon real-world tasks, 2026. URL https://arxiv.org/abs/2606.29537.

Z.AI. GLM-5.1. https://docs.z.ai/guides/llm/glm-5.1, 2026. Flagship model for long-horizon agentic tasks, released April 2026.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. AgentTuning: Enabling generalized agent abilities for LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3053–3077. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.181. URL https://aclanthology.org/2024.findings-acl.181/.

Weihuang Zheng, Tianyuan Zou, Eileen Ye, Alphet Liu, Youyong Kong, Ya-Qin Zhang, Duran Zheng, and Maxm Pan. E-Bench: Benchmarking multi-step tool-use agents in real-world product scenarios, 2026. URL https://arxiv.org/abs/2607.23722.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, et al. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations (ICLR)*, 2024.

Yang Zhou, Hongyi Liu, Zhuoming Chen, Yuandong Tian, and Beidi Chen. GSM-Infinite: How do your LLMs behave over infinitely increasing context length and reasoning complexity? *arXiv preprint arXiv:2502.05252*, 2025.