STARTUPBENCH: BENCHMARKING GENERAL-PURPOSE AGENTS ON MARKET-VALIDATED END-TO-END WORKFLOWS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in Large Language Models(LLMs) and agents have substantially improved the ability of AI systems to execute complex tasks. Yet existing benchmarks largely rely on researcher-selected tasks, leaving uncertain whether such progress extends to the work that real-world users actually demand from AI systems. We introduce **StartupBench**, an E2E agent benchmark grounded in market-validated AI startup products. Rather than defining tasks from predefined assumptions about useful agent capabilities, we systematically study AI products with demonstrated adoption, together with their product workflows and users, to identify real-world tasks for which AI has established practical demand across diverse professional domains. We translate these workflows into complete deliverable-oriented tasks and evaluate them with fine-grained rubrics capturing their complex requirements. Across representative models evaluated under a unified agent harness, even the strongest model successfully completes only approximately 30% of StartupBench, despite making substantial partial progress on many tasks. Further analysis identifies aspects like complex instruction following and domain-specific expertise as major sources of failure. Our results reveal that many market-validated workflows remain beyond the reliable capabilities of current general-purpose agents, establishing StartupBench as an empirical measure of progress toward E2E completions of real-world user tasks. Our data is available at `https://anonymous.4open.science/r/Startup_bench-1CB6`.

1 INTRODUCTION

Large language models (LLMs) are increasingly evolving from systems that primarily retrieve information or answer questions (Du et al., 2026; Mialon et al., 2024; Wang et al., 2024) into agents capable of completing complex real-world tasks. Current AI systems are expected to execute end-to-end (E2E) workflows, coordinate multiple capabilities, and produce deliverables that satisfy practical requirements under realistic constraints. As AI becomes increasingly integrated into professional work, autonomously completing user-delegated tasks has become an important measure of model capability. Evaluating whether AI systems can reliably translate their capabilities into professionally usable deliverables is therefore a central challenge for the next generation of benchmarks.

Recent benchmarks have made important progress toward evaluating E2E task execution under realistic settings. Nevertheless, several important limitations remain. First, many benchmark tasks are still largely defined from researchers' perspectives rather than grounded in workflows that have demonstrated practical demand through real-world AI adoption (Phan et al., 2025; Patwardhan et al., 2026; Mialon et al., 2024), limiting their ability to reflect authentic user needs and deliverable requirements. Second, although several benchmarks evaluate complete deliverables (Tang et al., 2026; Sun et al., 2026), their evaluation protocols are often substantially coarser than the deliverables themselves. Real-world work products typically involve numerous functional, structural, formatting, and domain-specific requirements that holistic evaluations cannot faithfully assess. Consequently, existing benchmarks still provide only limited evidence of *how well current models and agents can satisfy complex real-world user requirements and reliably produce professionally usable deliverables*.

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.

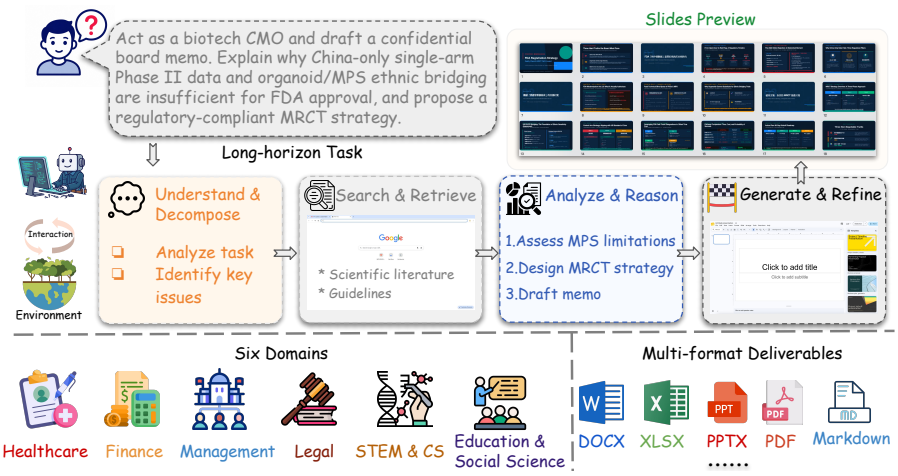


Figure 1: Overview of StartupBench: real AI-product workflows, long-horizon agent execution, multiple-domain coverage, and multi-format deliverable generation.

To address these limitations, we observe that AI-native startups offer a natural source of real-world AI workflows: their products have been validated through real-world adoption and commercial demand, providing direct evidence that users value and seek to delegate these tasks to AI. We therefore introduce **StartupBench**, a survey- and interview-driven benchmark that draws on these workflows to capture realistic E2E user requirements across domains. StartupBench spans 97 tasks across 6 primary domains and is built around the following design principles:

- **Adoption-grounded task sourcing.** We systematically survey AI-native startups and their products, complemented by product demonstrations and user interviews, to identify workflows with demonstrated real-world demand. We then construct these workflows into benchmark tasks that preserve their core user objectives and practical requirements.
- **E2E deliverables.** Each task requires producing the complete deliverables that users ultimately expect in realistic workflows, rather than intermediate outputs or simplified demonstrations.
- **Fine-grained rubric-based evaluation.** Each StartupBench task is evaluated using multiple independent rubrics covering distinct dimensions of the expected deliverable, including functionality, structure, formatting, and domain-specific quality, ensuring a comprehensive and faithful assessment of real-world task completion.

We evaluate representative models on StartupBench under unified agentic harness. Results show that even the strongest model successfully completes only about 30% of benchmark tasks, indicating that current general-purpose systems remain far from reliably producing professionally usable deliverables. Performance varies substantially across domains, with Finance, STEM & Computer Science and Education & Humanities proving particularly challenging. Deeper analysis further attributes these failures primarily to limitations in complex instruction following and domain-specific expertise, highlighting promising directions for future foundation models and agent systems.

To summarize, our contributions are as follows:

- We propose a survey- and interview-driven methodology for constructing agent benchmarks from real AI adoption scenarios.
- We construct StartupBench, a benchmark built through AI-native startup research, enterprise-user interviews, user-oriented task abstraction, expert data construction, and multi-stage quality control.
- We present a comprehensive empirical study of current foundation models and general-purpose agents on startup-derived workflows, providing a practical evaluation signal for assessing whether AI agents are progressing from answering questions toward reliably completing real work.

Table 1: Comparison of StartupBench with existing benchmarks. E2E denotes end-to-end workflow completion; Item-wise Eval. denotes criterion-level judging. ✓, ✗, and △indicate full, no, and partial support.

Benchmark	Task Source	Final Output	E2E	Item-wise Eval.	Process Logic	Open-ended	Evaluation Method
GAIA (Mialon et al., 2024)	Manual	Text	△	✗	✗	✗	Rule
OSWorld (Xie et al., 2024)	Real+Manual	Env State	△	✗	✗	✗	Execution
DAComp (Lei et al., 2026)	Real+Manual	Workspace	✓	✗	✗	✓	Hybrid
GDPval (Patwardhan et al., 2026)	Manual	Work products	✓	✗	✗	△	Manual
Workspace-Bench (Tang et al., 2026)	Sim.+Manual	Workspace	✓	✗	✗	✓	Agent-as-Judge
OneMillionBench (Yang et al., 2026)	Expert-authored	Text	△	✗	△	✓	LLM-as-a-judge
Agents' Last Exam (Sun et al., 2026)	Occupation-guided	Workspace	✓	✗	△	△	Hybrid
OfficeQA Pro (Opsahl-Ong et al., 2026)	Corpus-grounded	Text	✗	✗	✗	✗	Answer match
StartupBench (Ours)	**Market-vetted + Expert-built**	**Workspace**	✓	✓	✓	✓	**Agent-as-a-judge**

2 Related Work

2.1 Commercial AI agents and vertical workflows

Agents now support planning, tool use, file processing, and multi-step execution (Yao et al., 2023; Schick et al., 2023), enabling commercial agents for document processing, data analysis, financial research, enterprise operations, and knowledge work. These products show that practical value depends on packaging model capabilities into useful work products, yet academic benchmarks rarely use commercial adoption for task discovery.

2.2 End-to-end Agent Evaluation

Agent benchmarks have progressed from tool use, multi-step reasoning, and general assistance (AgentBench and GAIA) (Liu et al., 2024; Mialon et al., 2024), through web and computer-use environments (Zhou et al., 2024; Xie et al., 2024), to enterprise software, simulated workplaces, occupational tasks, and long-horizon professional workflows (Drouin et al., 2024; Xu et al., 2026; Patwardhan et al., 2026; Hu et al., 2026; Zhu et al., 2026; Meng et al., 2026). StartupBench shares this E2E focus but sources tasks from AI-native products, ToB user needs, and expected business deliverables rather than public environments, occupational taxonomies, or simulations.

2.3 Agent-as-a-Judge for Complex Deliverables

As outputs expand from answers to reports, spreadsheets, slides, code patches, and structured files, evaluation must become deliverable-centric (You et al., 2026). MT-Bench and Prometheus study preference- or rubric-conditioned LLM judges (Zheng et al., 2023; Kim et al., 2024), but agent outputs also require evidence retrieval, file inspection, state verification, and constraint checking. Agent-as-a-Judge adds tools and environment interaction, while AJ-Bench evaluates information acquisition, state verification, and process verification (Zhuge et al., 2025; Shi et al., 2026).

3 StartupBench

StartupBench aims to evaluate whether models and agents can complete realistic workflows that users already delegate to AI products. This requires tasks that reflect authentic user demand while remaining answerable, evaluable, and sufficiently challenging to distinguish current models and agents. However, manually designed tasks may lack real-world grounding, while directly extracted product demonstrations often lack the context and specifications needed for reproducible evaluation. To bridge this gap, we adopt a multi-stage construction pipeline shown in Figure 2, surveying market-validated AI-native startups, interviewing deep users to identify representative workflows, engaging domain experts to construct benchmark tasks, and applying rigorous quality control to ensure task validity and discriminative difficulty.

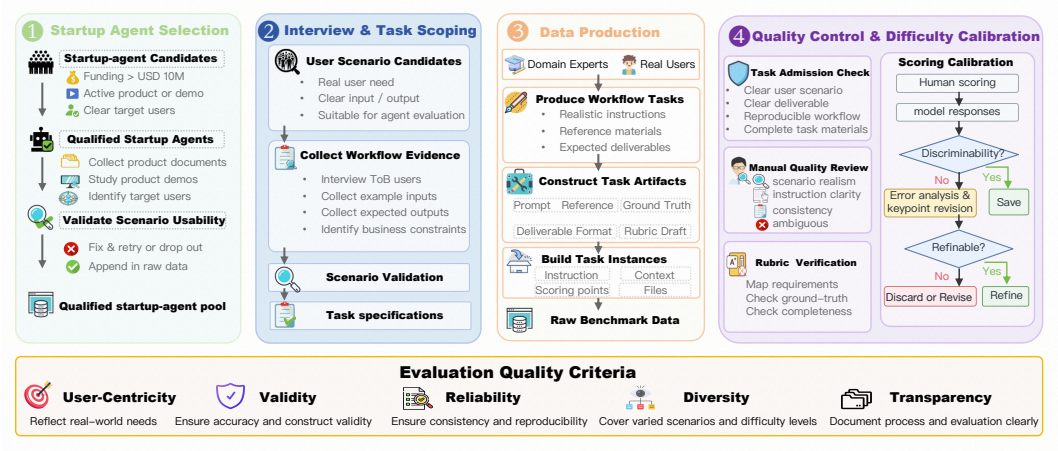


Figure 2: Construction pipeline of StartupBench.

3.1 Dataset Construction

3.1.1 Startup Survey for Market-Validated Agent Scenarios

We first survey AI-native startups that build agent products across diverse domains. To identify workflows with market validation, we retain startups with more than USD 10M in funding and require evidence of real adoption, either through paid usage or substantial user traction. Funding provides a coarse signal of market expectation, while payment or user-scale evidence indicates that the product is used beyond exploratory demonstrations. The output of this stage is a pool of candidate products and workflows for user interviews. Detailed criteria for startup agent selection is shown in Appendix A. During this stage we collect 20+ start-up agents as the basis for subsequent stages.

3.1.2 User Interviews for Scenario and Requirement Discovery

For each candidate startup product, we interview deep users of the corresponding agent from different domains to recover practical usage beyond public product descriptions. The interviews elicit the usage context, user goal, input information, expected deliverable, success criteria, and common constraints. We summarize these findings into scenario specifications and representative demo cases, which serve as anchors for following construction stages. During this stage we interview over 30 deep users and collect at least one demo case for each candidate Agent.

3.1.3 Expert Construction of Domain Tasks

Given the interview-derived workflow specifications and representative user scenarios as references and seed tasks, we recruit over 50 domain experts to reconstruct benchmark tasks that preserve the original workflow objectives, practical constraints, and expected deliverables while adapting them into reproducible evaluation instances. Rather than designing synthetic problems, experts standardize authentic user requests into benchmark tasks that faithfully represent real professional workflows, with **each task grounded in a representative use case of a corresponding startup agent**. Every task is required to satisfy three principles: it must originate from a realistic work request that users would naturally delegate to AI agents, involve an end-to-end workflow rather than an isolated subtask, and produce concrete professional deliverables with clearly defined success criteria and fine-grained evaluation rubrics for objective evaluation.

To standardize these tasks for evaluation, each task is represented as a triple

$$\mathcal{T} = (q, \mathcal{E}, \mathcal{R}),$$

where q is a natural-language user request describing the task, $\mathcal{E}$ denotes the workspace containing all input files and resources required to complete the task, and $\mathcal{R} = \{(p_i, w_i)\}_{i=1}^{n}$ is a set of weighted evaluation rubrics that assess complementary aspects of deliverable quality. Detailed task

Table 2: Task distribution of StartupBench.

Domain	Number of tasks
Medical & Healthcare	21
Finance	18
Legal	16
Business & Management	19
STEM & Computer Science	16
Education & Humanities	7

specifications and annotation guidelines are provided in Appendix B.1. During this stage, we collect over 150 candidate tasks for further quality control.

3.1.4 Quality Control

To ensure benchmark quality and consistency, every task undergoes the following multi-stage quality control process:

Expert Cross Validation. Every task is independently reviewed by at least one additional domain expert. Reviewers verify the following aspects of task quality: (1) the authenticity and correctness of the task description and workspace, (2) the fidelity of the reconstructed workflow, (3) the validity of the evaluation rubrics, and (4) the consistency between the reference deliverable and the finalized evaluation protocol. Tasks failing to meet any of these criteria are returned to the original expert for revision and re-evaluation, with those failing to satisfy the requirements after revision excluded from the benchmark. Over 30 tasks were excluded at this stage due to quality issues. Detailed review criteria are provided in Appendix B.2.

Difficulty Control. We calibrate task difficulty through pilot executions using several frontier models under the same evaluation harness adopted in the benchmark[1]. Tasks that are consistently completed with high quality are excluded due to limited discriminative value. In total, approximately 30 tasks are excluded through this difficulty-control process. Pilot results also serve as an additional quality assurance stage: when failures are attributed to ambiguous instructions, incomplete workspace information, or inadequately specified evaluation criteria rather than intrinsic task difficulty, the corresponding task is revised and re-validated before inclusion.

3.2 Statistics for StartupBench

StartupBench contains 97 real-world workflow tasks across 6 top-level domain, derived from 10 distinct startup agents (see Appendix D for details). These tasks are designed to reflect deliverable-oriented workflows in realistic office settings, requiring agents to understand task contexts, follow complex constraints, process heterogeneous files, and produce usable work products. For evaluation, each task is evaluated using an average of **25.3** fine-grained rubrics, spanning **6 dimensions and 3 importance levels**, enabling detailed assessment of complex, long-horizon workflows and diverse work deliverables. The summary of the domain coverage of StartupBench is shown in Table 2.

3.3 Automatic Evaluation of StartupBench Tasks

As stated above, every task in StartupBench is defined as a triple $\mathcal{T} = (q, \mathcal{E}, \mathcal{R})$. Given $(q, \mathcal{E})$, the evaluated model produces a set of deliverables $\mathcal{D}$. Before evaluation, we construct a lightweight evidence view $\mathcal{V}(\mathcal{D})$, consisting of extracted textual content and rendered page images, to facilitate navigation over heterogeneous artifacts while allowing to access the original files if necessary.

Rather than relying on a single holistic judgment, StartupBench evaluates each rubric item independently through a dedicated AgentJudge session. Most tasks contain 20+ fine-grained rubric items

[1]We randomly select models from GPT-5.5, Seed-2.1-Pro and GLM-5.1 as the model for pilot executions, intending to construct a challenging and discriminative benchmark rather than to estimate performance over the natural task distribution.

covering diverse aspects of the expected deliverables, including functionality, completeness, formatting, and domain-specific requirements. Evaluating these criteria separately allows each rubric to receive dedicated reasoning and tool usage without interference from unrelated evaluation dimensions, while ensuring that every requirement is assessed explicitly. The final task score is then obtained by aggregating the outcomes of all rubric-level evaluations. Then following **Agent-as-a-Judge**, each rubric evaluation is carried out by a lightweight agent rather than a single forward pass. Let J denote the judging prompt and evaluation protocol used by the AgentJudge, given the task specification, deliverables, evidence view, and the target rubric item p_i, the judge produces both a binary decision $y_i \in \{0, 1\}$ indicating whether the rubric is satisfied and a textual justification e_i:

$$(y_i, e_i) = \text{AgentJudge}\big(J, q, \mathcal{D}, \mathcal{V}(\mathcal{D}), p_i\big), \tag{1}$$

The final task score is computed by aggregating the weighted decisions across all rubric items:

$$\text{score}(\mathcal{D}, \mathcal{T}) = \frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i} \in [0, 1]. \tag{2}$$

4 Experiments

4.1 Experimental Setup

We evaluate 9 representative models on StartupBench, including: GPT-5.6-sol (OpenAI, 2026), GPT-5.5 (OpenAI, 2026), Gemini-3.1-Pro (Google Cloud, 2026), Seed-2.1-Pro (ByteDance Seed, 2026), Qwen-3.6-Max (Qwen Team, 2026), DeepSeek-V4-Pro (DeepSeek-AI, 2026), Kimi-K3 (Team et al., 2026), Kimi-K2.6 (Moonshot AI, 2026), and GLM-5.1 (Zeng et al., 2026). All models are evaluated using the same Nanobot harness (HKUDS, 2026) and tool configuration, with a maximum of **200** interaction steps per task. Each task is run independently **3** times, with the agent initialized from the task-specific workspace and input query. The resulting deliverables are evaluated against the task specification, workspace context, and rubric set by a judge agent using GPT-5.5 under the same harness. We report the mean task score, together with the **success rate** pooled across the runs. A run is considered successful when its final score is at least **90%**.

4.2 Main Results

Startup tasks remain far from saturated under a general-purpose agent harness. As shown in Table 3, Kimi-K3 and GPT-5.6-sol achieve average scores of 73.67 and 73.61, respectively, yet no model successfully completes even one third of the benchmark under the strict acceptance criterion (score $\geq$ 90). Notably, Kimi-K3 achieves a slightly higher average score than GPT-5.6-sol but a lower success rate (29.55% vs. 31.27%), highlighting the discrepancy between satisfying individual task requirements and successfully completing entire workflows. More broadly, the substantial gap between average scores and success rates across all models suggests that current agents can make considerable progress on professional tasks but still struggle to consistently deliver artifacts meeting professional standards. We further investigate the underlying causes of this gap in Section 4.4.2.

Performance varies substantially across professional domains. Table 3 shows that Business is consistently easiest: every model scores above 60 on average, and its mean success rate is the highest across domains. Finance is hardest, with the lowest average score (54.48%), suggesting that structured, standardized tasks are easier than workflows requiring sustained quantitative reasoning, cross-document consistency, and multi-step verification. The nearly tied leaders (73.67 vs. 73.61 overall) also specialize differently. By success rate, Kimi-K3 leads Medical and Business, GPT-5.6-sol leads Finance, STEM, and Education, and GPT-5.5 leads Legal; by average score, Kimi-K3 leads Medical, Business, Finance, and Education, whereas GPT-5.6-sol leads Legal and STEM. No model therefore dominates all domains, exposing substantial domain-dependent variation and room to improve robust cross-domain transfer.

4.3 Effectiveness of Proposed Evaluation Framework

To validate the reliability of our evaluation framework, we sample one trial from each of two representative models, Seed-2.1-Pro and GPT-5.5, across all 97 StartupBench tasks, yielding 194 trials and 4,906 rubric-level judgments for expert validation. As shown in Appendix F, our automatic

Table 3: StartupBench leaderboard. Top: average score, with 95% bootstrap confidence intervals within 3 trials; bottom: success rate (score $\geq$ 90%). Best and second-best values per column are bold and underlined.

Model	Overall	Med	Finance	Legal	Business	STEM	Education
Average score							
Kimi-K3	**73.67** [71.74, 75.45]	**80.03**	**62.38**	69.45	**83.90**	68.34	**77.71**
GPT-5.6-sol	<u>73.61</u> [71.82, 75.37]	<u>79.32</u>	61.66	**73.53**	77.61	**74.77**	<u>73.94</u>
GPT-5.5	72.79 [71.25, 74.24]	78.59	<u>62.09</u>	<u>71.75</u>	<u>79.59</u>	<u>72.14</u>	68.32
Seed-2.1-Pro	67.19 [65.65, 69.10]	73.15	57.26	61.35	77.83	65.61	62.91
GLM-5.1	60.77 [58.50, 63.47]	65.66	51.22	50.20	78.53	52.18	66.50
Kimi-K2.6	59.95 [57.49, 62.25]	67.70	51.17	54.62	71.91	51.38	58.60
Qwen-3.6-Max	59.46 [57.71, 61.16]	66.51	52.45	50.33	75.03	48.83	59.26
DeepSeek-V4-Pro	61.11 [59.54, 62.63]	69.42	51.11	54.53	73.69	54.89	57.03
Gemini-3.1-Pro	49.73 [47.46, 51.96]	54.05	41.00	43.49	61.18	45.88	51.23
Success rate ($\geq$ 90)							
Kimi-K3	<u>29.55</u>	**34.92**	20.37	20.83	**52.63**	16.67	<u>23.81</u>
GPT-5.6-sol	**31.27**	<u>33.33</u>	**27.78**	<u>22.92</u>	38.60	**33.33**	**28.57**
GPT-5.5	26.80	30.16	<u>22.22</u>	**25.00**	38.60	<u>22.92</u>	9.52
Seed-2.1-Pro	22.34	19.05	16.67	14.58	<u>45.61</u>	20.83	4.76
GLM-5.1	16.49	12.70	16.67	8.33	36.84	8.33	9.52
Kimi-K2.6	13.06	15.87	14.81	6.25	24.56	4.17	4.76
Qwen-3.6-Max	15.12	11.11	16.67	10.42	33.33	6.25	4.76
DeepSeek-V4-Pro	16.49	20.63	16.67	8.33	29.82	10.42	0.00
Gemini-3.1-Pro	6.53	6.35	11.11	6.25	8.77	0.00	4.76

evaluator achieves an overall rubric-level accuracy of **91.91%**, with a precision of **97.19%**, recall of **90.90%**, and F1 score of **93.94%** against expert annotations. Agreement remains consistently high across both models, including GPT-5.5, which also serves as the underlying judge model. At the task level, the evaluator achieves **93.30%** agreement with expert success decisions under the score threshold of 90. These results demonstrate substantial consistency between automated and expert evaluations at both the rubric and task levels, supporting the reliability of our rubric-based evaluation framework.

We further investigate the importance of rubric-wise evaluation by comparing it with a holistic setting, where a single judge agent evaluates all rubrics in one session. This reduces agreement with human experts to 83% and introduces substantial execution instability, with malformed or incomplete outputs requiring over 2 runs on average to obtain a valid evaluation. In contrast, independently evaluating each rubric improves both agreement with human experts and execution robustness.

4.4 Failure-Mode Analysis

4.4.1 Outcome-level Failure Analysis

Capability differences across rubric dimensions. Appendix J further breaks down model performance by rubric dimension. Overall, current models achieve consistently higher scores on Structure & Completeness, Information Integration, and especially Output & Presentation, indicating that they are generally capable of organizing heterogeneous information into well-structured and visually polished deliverables. In contrast, Domain-Specific Compliance emerges as the most challenging dimension across all evaluated models, while Calculation Precision also remains substantially weaker than the other dimensions. This suggests that although modern foundation models can produce artifacts that appear complete and professional, reliably satisfying domain-specific regulations, business conventions, and numerical correctness continues to be a major obstacle for real-world deployment.

The two strongest models, GPT-5.6-sol and Kimi-K3, outperform the remaining systems across nearly all rubric dimensions rather than relying on a single specialized capability. Their largest

advantages are observed in Structure & Completeness, Calculation Precision, Information Integration, and Output & Presentation, demonstrating stronger end-to-end workflow execution and artifact construction capabilities. Nevertheless, even these leading models obtain their lowest scores on Domain-Specific Compliance, indicating that adherence to professional standards and domain-specific requirements remains the primary bottleneck even for state-of-the-art systems.

Models often satisfy peripheral requirements while missing what matters most. Table 11 reports task-normalized rubric satisfaction rates across the three importance levels. Averaged across models, satisfaction decreases from **68.67%** for Auxiliary rubrics to **65.89%** for Important rubrics and **63.45%** for Core rubrics, with 8 of the 9 models exhibiting the same overall trend. Importantly, these levels reflect each requirement's contribution to successful task completion rather than its expected difficulty. This pattern reveals a characteristic failure mode of current agents: they can often fulfill auxiliary or surface-level requirements while failing on the core requirements determining whether the task is actually completed successfully. In other words, substantial apparent progress can be driven by satisfying less consequential parts of the task, while critical omissions still prevent the final deliverable from being practically usable. Appendix J provides a typical case.

Output format non-compliance. Even this deterministic requirement, involving neither complex reasoning nor domain knowledge, remains imperfect on the 56 tasks with explicit output-file rubrics. As shown in Appendix H, compliance ranges from 86.3% to 97.6%. Violations mainly use the wrong type (e.g., `.md`/`.txt` instead of `.pdf`) or omit a requested deliverable (e.g., producing only one of `.xlsx` and `.docx`), showing that elementary output constraints remain a non-negligible failure source.

4.4.2 Behavior-level Failure Analysis

Complex instruction following. Real workflows combine structural, computational, formatting, and executable constraints. Models often satisfy the most visible subset while missing equally critical conditions, producing artifacts that appear complete but fail direct-use acceptance criteria. This suggests optimization for apparent correctness rather than complete instruction execution. Appendix K.1 presents an HR-payroll example.

Self-verification hallucination. Many models conflate successful execution with verified delivery: they summarize their reasoning or completed steps without checking the final artifact against the request. Wrong cell values, unmatched identifiers, missing constraints, and formatting inconsistencies therefore survive in otherwise plausible outputs. This self-verification hallucination is deployment-critical because office workflows require artifact-level validation, not confidence in the execution process; Appendix K.2 gives an example.

Domain-specific expertise. Another major failure source, consistent with weak Domain-Specific Compliance, is that fluent, professional-looking outputs miss domain-specific operational requirements. The pattern varies by domain. Business tasks stress spreadsheet, dashboard, formula, and report correctness. Finance stresses attribution, numerical precision, valuation assumptions, temporal consistency, reconciliation, and accounting conventions. Legal failures involve incomplete or incorrect reasoning, citations, factual coverage, statutory support, or fact-to-rule mapping. Medical scores are relatively higher, but errors in medication timing, continuation criteria, or discharge planning create greater safety risks. STEM and Computer Science require precise grounding among code, drawings, images, bills of materials, and system components. Education and Humanities emphasize detailed rule following, including curricula, credits, document organization, and textual coherence. Appendix K.3 illustrates insufficient clinical actionability.

4.5 Impact of Harness on Startup Tasks

4.5.1 Effect of General-Purpose Harness

In the main experiment, all models are evaluated under nanobot Agent framework. To examine whether the observed performance is primarily determined by the choice of agent framework rather than the underlying model, we re-evaluate the benchmark using two more representative general-purpose agent frameworks: Hermes (Nous Research, 2026) and Claude Code (Anthropic). All frameworks are evaluated under the same task set and identical evaluation protocol, with only the execution framework replaced.

Table 4: Performance under different general-purpose agent frameworks.

Model	Hermes	Claude Code	Nanobot	Max–Min
GPT-5.5	71.00	70.11	72.79	2.68
GLM-5.1	60.20	60.18	60.77	0.59
Qwen-3.6-Max	59.10	57.38	59.46	2.08
Average	63.43	62.56	64.34	1.78

Table 5: General-purpose versus specialized agent systems. Oracle General denotes the averages of the highest score achieved in all trials for every model.

Agent System	Avg Score	Success Rate
General (All Runs avg)	64.31	19.74
General (Oracle)	71.75	28.06
Specialized Agent	**83.50**	**39.18**

As shown in Table 4, replacing the general-purpose framework leads to only minor performance differences. Across all evaluated models, the average variation between the best and worst framework is only 1.78 points. GLM-5.1 is almost completely unaffected (0.59 points), while even the largest difference observed on GPT-5.5 is only 2.68 points. More importantly, the relative ordering of models remains unchanged across all three frameworks. These results suggest that StartupBench is largely robust to the choice of general-purpose agent framework. Although different frameworks adopt different prompting strategies, tool interfaces, and interaction mechanisms, simply replacing one general-purpose framework does not substantially alter end-to-end task performance. Consequently, the performance gap observed on StartupBench primarily reflects the capability of the underlying foundation models rather than implementation-specific characteristics of a particular framework.

4.5.2 General-Purpose Agents versus Specialized Agents

We compare general-purpose agents with the specialized startup agents from which StartupBench tasks are derived, evaluating each task using its corresponding production agent. To account for execution variability, we additionally consider an oracle setting for general-purpose agents, retaining the best outcome across three independent trials for each model–task pair.

As shown in Table 5, specialized agents achieve an average score of 83.50 and a success rate of 39.18%, compared with 64.31 and 19.74% for general-purpose agents. Even under the oracle setting, general-purpose agents achieve only 71.75 and 28.06%, leaving gaps of 11.75 points in average score and 11.12 percentage points in success rate. These results highlight the substantial advantage of specialized agent systems on their target workflows, suggesting that stronger foundation models alone may not fully bridge the gap without improvements in domain-specific knowledge, tool orchestration, and workflow execution.

5 Conclusion

We introduce **StartupBench**, a benchmark for evaluating models on end-to-end professional workflows derived from market-validated AI-native products and real-world user demands. Across multiple tasks spanning diverse domains, our evaluation reveals a substantial gap between making meaningful progress on professional work and producing deliverables that fully satisfy practical acceptance criteria. This gap is primarily driven by limitations in complex instruction following, domain-specific expertise, professional conventions, and long-horizon workflow execution. By grounding evaluation in workflows that users already seek to delegate to AI, StartupBench provides a realistic measure of practical agent capabilities and highlights concrete directions for future improvement. We hope StartupBench supports the development of models and agents that can not only perform valuable professional work, but reliably deliver results ready for direct use.

AI USE STATEMENT

Generative AI tools were used to assist with LaTeX template migration, formatting diagnostics, and language compression. All AI-assisted edits were reviewed by the authors, who take responsibility for the final manuscript content.

REPRODUCIBILITY STATEMENT

We make several efforts to ensure the reproducibility of our benchmark and experimental results. The benchmark data are available through the anonymous link provided in the abstract. We document the benchmark construction and evaluation protocols in detail, and report major experimental settings in detail, including the evaluated models, agent harnesses, judge models, and scoring procedures, in the main text and appendices. These resources and specifications are intended to enable reproduction of both the benchmark evaluation and the analyses presented in this work.

REFERENCES

Anthropic. Claude code. https://github.com/anthropics/claude-code. GitHub repository.

ByteDance Seed. Seed2.1 model card: Agentic intelligence for productivity. Technical report, ByteDance, 2026. URL https://lf3-static.bytednsdoc.com/obj/eden-cn/lapzild-tss/ljhwZthlaukjlkulzlp/seed2.1/Seed2_1_Model_Card.pdf.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://huggingface.co/collections/deepseek-ai/deepseek-v4. Technical Report.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, Léo Boisvert, Megh Thakkar, Quentin Cappart, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. Workarena: How capable are web agents at solving common knowledge work tasks?, 2024. URL https://arxiv.org/abs/2403.07718.

Xeron Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *Advances in Neural Information Processing Systems*, 38, 2026.

Google Cloud. Gemini 3.1 Pro Model Card, 4 2026. URL https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/3-1-pro?hl=zh-cn.

HKUDS. Nanobot: An open-source ultra-lightweight ai agent framework. https://github.com/HKUDS/nanobot, 2026. Accessed: 2026-06-13.

Xiaomeng Hu, Yinger Zhang, Fei Huang, Jianhong Tu, Yang Su, Lianghao Deng, Yuxuan Liu, Yantao Liu, Dayiheng Liu, and Tsung-Yi Ho. Occubench: Evaluating ai agents on real-world professional tasks via language environment simulation, 2026. URL https://arxiv.org/abs/2604.10866.

Seungone Kim, Jay Shin, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Ryan Shin, Sungdong Kim, James Thorne, Minjoon Seo, et al. Prometheus: Inducing fine-grained evaluation capability in language models. In *International Conference on Learning Representations*, volume 2024, pp. 29927–29962, 2024.

Fangyu Lei, Jinxiang Meng, Yiming Huang, Yitong Zhang, Jianwen Luo, Xin Zou, Ruiyi Yang, Wenbo Shi, Yan Gao, Shizhu He, et al. Dacomp: Benchmarking data agents across the full data intelligence lifecycle. In *International Conference on Learning Representations*, volume 2026, pp. 104463–104501, 2026.

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*, 2024.

Jinxiang Meng, Shaoping Huang, Fangyu Lei, Jingyu Guo, Haoxiang Liu, Jiahao Su, Sihan Wang, Yao Wang, Enrui Wang, Ye Yang, et al. Dv-world: Benchmarking data visualization agents in real-world scenarios. *arXiv preprint arXiv:2604.25914*, 2026.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *International Conference on Learning Representations*, 2024.

Moonshot AI. Kimi K2.6: Advancing open-source coding. https://www.kimi.com/blog/kimi-k2-6, 2026. Accessed: 2026-06-02.

Nous Research. Hermes agent: The ai agent that learns from you. https://github.com/hermes-agent-org/hermes, 2026. GitHub repository.

OpenAI. Gpt-5.6 system card. Technical report, OpenAI, 7 2026. URL https://deploymentsafety.openai.com/gpt-5-6/gpt-5-6.pdf.

OpenAI. Introducing gpt-5.5, 4 2026. URL https://openai.com/index/introducing-gpt-5-5/. System Card: https://deploymentsafety.openai.com/gpt-5-5/gpt-5-5.pdf.

Krista Opsahl-Ong, Arnav Singhvi, Jasmine Collins, Ivan Zhou, Cindy Wang, Ashutosh Baheti, Owen Oertell, Jacob Portes, Sam Havens, Erich Elsen, et al. Officeqa pro: An enterprise benchmark for end-to-end grounded reasoning. *arXiv preprint arXiv:2603.08655*, 2026.

Tejal Patwardhan, Rachel Dias, Elizabeth Proehl, Grace Kim, Michele Wang, Olivia Watkins, Simon Fishman, Marwan Aljubeh, Phoebe Thacker, Laurance Fauconnet, et al. Gdpval: Evaluating ai model performance on real-world economically valuable tasks. In *International Conference on Learning Representations*, volume 2026, pp. 24005–24040, 2026.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

Qwen Team. Qwen3.6-Max-Preview: Smarter, sharper, still evolving, April 2026. URL https://qwen.ai/blog?id=qwen3.6-max-preview.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

Wentao Shi, Yu Wang, Yuyang Zhao, Yuxin Chen, Fuli Feng, Xueyuan Hao, Xi Su, Qi Gu, Hui Su, Xunliang Cai, et al. Aj-bench: Benchmarking agent-as-a-judge for environment-aware evaluation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 25371–25413, 2026.

Yiyou Sun, Xinyang Han, Weichen Zhang, Yuanbo Pang, Tianyu Wang, Yuhan Cao, Yixiao Huang, Chris Duroiu, Haoyun Zhang, Jeffrey Lin, et al. Agents' last exam. *arXiv preprint arXiv:2606.05405*, 2026.

Zirui Tang, Xuanhe Zhou, Yumou Liu, Linchun Li, Yukai Wu, Weizheng Wang, Hongzhang Huang, Wei Zhou, Jun Zhou, Jiachen Song, et al. Workspace-bench 1.0: Benchmarking ai agents on workspace tasks with large-scale file dependencies. *arXiv preprint arXiv:2605.03596*, 2026.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

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

Frank Fangzheng Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, et al. Theagentcompany: benchmarking llm agents on consequential real world tasks. *Advances in Neural Information Processing Systems*, 38, 2026.

Qianyu Yang, Yang Liu, Jiaqi Li, Jun Bai, Hao Chen, Kaiyuan Chen, Tiliang Duan, Jiayun Dong, Xiaobo Hu, Zixia Jia, Yang Liu, Tao Peng, Yixin Ren, Ran Tian, Zaiyuan Wang, Yanglihong Xiao, Gang Yao, Lingyue Yin, Ge Zhang, Chun Zhang, Jianpeng Jiao, Zilong Zheng, and Yuan Gong. $onemillion-bench: How far are language agents from human experts?, 2026. URL https://arxiv.org/abs/2603.07980.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

Runyang You, Hongru Cai, Caiqi Zhang, Qiancheng Xu, Meng Liu, Tiezheng Yu, Yongqi Li, and Wenjie Li. Agent-as-a-judge. *arXiv preprint arXiv:2601.05111*, 2026.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024.

Liya Zhu, Jingzhe Ding, Jian Zhang, Jianbo Xue, Shihao Liang, Ge Zhang, Yi Zhu, Duju Zeng, Xiang Gao, Qingshui Gu, Mailun Gao, Huimin Che, Yan Zhao, Peiheng Zhou, Haojun Wang, Chaobo Xian, Lili Le, Chi Wu, Yiwei Liu, Shengda Long, Jiale Yang, Fangzhi Xu, Sijin Wu, Haodong Duan, Chao He, Zhaojian Li, Minchao Wang, Huan Zhou, Jiani Hou, Chuqian Yu, Weiran Shi, Hongwan Gao, Jiamin Chen, Guanhong Chen, Tingqin Luo, Kaiyuan Zhang, Zhixin Yao, Qing Hua, Yuhao Jiang, Jin Chen, Pu Chen, Zhenyu Hu, Xingyu Li, Zhengxuan Jiang, Meng Cao, Tianfeng Long, Haozhe Wang, Mingzhang Wang, Yichen Zhang, Yiming Dai, Chenchen Zhang, Jiaying Wang, Xinying Liu, Xingzu Liu, Lingling Zhang, Xinjie Chen, Yujia Qin, Wangchunshu Zhou, Zhiyong Wu, Yang Liu, Jiaheng Liu, Lei Zhang, Shen Yan, Wenhao Huang, Zaiyuan Wang, and Xiaolong Chang. Workflow-gym: Towards long-horizon evaluation of computer-use agentic tasks in real-world professional fields, 2026. URL https://arxiv.org/abs/2606.11042.

Mingchen Zhuge, Changsheng Zhao, Dylan R. Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, Yangyang Shi, Vikas Chandra, and Jürgen Schmidhuber. Agent-as-a-judge: Evaluate agents with agents. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Nn9POI9Ekt.

A STARTUP AGENT SELECTION CRITERIA

To ensure the selected startup agents reflect real user deamnds and workflows, we mainly adopt a two-stage screening process to select startup agents, combining mandatory eligibility requirements with evidence of real-world market adoption.

Mandatory Eligibility Criteria. Eligible startups must satisfy four requirements: (1) *Funding*: over USD 10 million in cumulative publicly disclosed funding, supported by verifiable sources; (2) *Operational status*: active operation at the time of the survey, with the target agent product publicly released; (3) *Product type*: an AI agent application supporting real-world user workflows, rather than a foundation model, standalone infrastructure, or agent development framework; and (4) *Agent capabilities*: the ability to execute multi-step workflows based on user goals through tool use, external environment interaction, or autonomous task execution. Only startups satisfying all four criteria proceed to the second stage.

Market Adoption Verification. We further require verifiable evidence of actual product adoption, including publicly disclosed customers, documented commercial deployments, reported revenue metrics, or active user statistics. We collect and cross-check such evidence using official company websites, public funding databases, and news reports, retaining only products with at least one identifiable source of adoption evidence. Given the diversity of business models and user populations across professional domains, we do not impose a uniform numerical threshold on adoption metrics. Instead, we require documented evidence of real-world product usage to distinguish market-adopted agents from conceptual demonstrations.

During subsequent task construction and quality control, we do not enforce the retention of any particular startup agent; agents whose associated tasks all fail to meet our quality and difficulty requirements are excluded from the final benchmark.

B TUTORIAL OF STARTUPBENCH TASK ANNOTATION

B.1 TASK CONSTRUCTION

A STARTUP task can be defined as a triple $\mathcal{T} = (q, \mathcal{E}, \mathcal{R})$. The experts are required to provide all the elements of a task:

- **Input query**: The query q is a natural-language statement of the task, specifying the user instruction that initiates the task. Derived from interview-elicited user workflows and authentic product-use scenarios, the input query is supposed to reflect realistic user intent, including task background, implicit constraints and expected deliverables.
- **Task workspace**: The environment $\mathcal{E}$ denotes the complete workspace provided to the agent, including the multimodal source files together with any additional resources and interaction setting required to complete the task. The task workspace provides a self-contained execution environment for the agent. Each workspace includes all necessary input artifacts (e.g., files, datasets, or structured resources) required to complete the task, and ensures that the agent has access to sufficient input information to solve the problem. This design enables reproducible evaluation under controlled conditions while preserving realistic workflow structure.
- **Evaluation rubrics**: The rubric $\mathcal{R} = \{(p_i, w_i)\}_{i=1}^{n}$ is a checklist within which every item consists of a natural-language scoring point p_i and a positive importance weight w_i reflecting its importance to successful task completion. The evaluation rubrics define a set of fine-grained criteria used to assess task completion quality. Due to the complexity and multi-faceted nature of real-world workflows, each task is evaluated using multiple rubric items covering different dimensions. We categorize all rubrics in StartupBench into 6 dimensions and 3 importance levels, which are detailed in Appendix E.

Together, these components preserve the structure of real-world agent workflows while enabling systematic and objective evaluation. Tasks may require information synthesis, constraint following, structured generation, or domain-specific reasoning and judgment.

Table 6: Summary of domain expert backgrounds. Professional experience refers to reported years of experience in the corresponding or closely related domain.

Expert Statistics	Value
Number of domain experts	57
Median professional experience	5 years
Experts with ≥ 3 years of experience	68.4%
Experts with ≥ 6 years of experience	42.1%
Experts with ≥ 10 years of experience	24.6%
StartupBench domain coverage	6 / 6

B.2 Cross Validation

To ensure reliability of the quality of data, in our construction process each task is independently reviewed by at least one additional domain expert with relevant professional experience. Rather than focusing only on annotation consistency, reviewers validate the realism and correctness of the task from multiple complementary perspectives.

- **Task authenticity.** Reviewers verify the authenticity of both the task description and the accompanying workspace. This includes checking that the input query reflects realistic user requests, that all provided materials are factually correct, and that no domain-specific knowledge errors exist.
- **Workflow fidelity.** Reviewers examine whether the task faithfully represents real-world workflows in the corresponding profession. Tasks that deviate from practical working procedures or fail to reflect genuine job responsibilities are revised or discarded.
- **Rubric validity.** Reviewers inspect the evaluation rubrics to ensure that the assessed capabilities are meaningful for the target workflow, that rubric items are clear and unambiguous, and that their relative weights reasonably reflect task priorities.
- **Ground-truth verification.** Reviewers validate the reference solution (ground truth) by evaluating it against the finalized rubrics. The ground-truth deliverable is required to achieve a full score. If inconsistencies are identified between the reference solution and the evaluation criteria, both the solution and the rubrics are revised until they are mutually consistent.

C Domain Expert Background

We recruit 57 domain experts to support task construction and cross-validation. The expert pool covers all 6 domains in StartupBench and spans a diverse range of professional backgrounds, including clinical medicine, law, investment and quantitative finance, software development and artificial intelligence, engineering, business management and consulting, and education and humanities. Experts are assigned to tasks according to their reported areas of expertise and professional experience, ensuring that task construction and review were conducted by individuals familiar with the corresponding professional domains.

Table 6 summarizes the professional experience of the expert pool. The median professional experience is 5 years. Among the 57 experts, 68.4% have at least 3 years of professional experience, 42.1% have at least 6 years, and 24.6% have at least 10 years. These backgrounds provide practical domain knowledge for translating interview-derived user scenarios into reproducible benchmark tasks and for assessing the realism, correctness, and evaluation criteria of tasks during cross-validation. All experts are reasonably compensated based on the actual workload associated with task construction and review.

D Selected Startup Agents in StartupBench

StartupBench comprises 97 professional tasks derived from the workflows of 10 selected AI agents across six domains. These agents include both domain-specific products and general-purpose agents

Table 7: Distribution of StartupBench tasks across professional domains and source agents. Numbers in parentheses indicate the number of tasks contributed by each agent within the corresponding domain. An agent may contribute tasks to multiple domains.

Domain	**# Tasks**	**Source Agents (# Tasks)**
Business & Management	19	Copilot (17), Perplexity AI (1), Alice (1)
Finance	18	Alice (7), Manus (4), Copilot (3), Perplexity AI (2), Gangtise AI (1), Fazhi (1)
STEM & Computer Science	16	Perplexity AI (11), Copilot (5)
Medical & Healthcare	21	Heidi Health (18), Copilot (2), OpenEvidence (1)
Legal	16	Fazhi (11), Manus (1), Spellbook (1), Perplexity AI (1), Copilot (1), Alpha AI (1)
Education & Humanities	7	Manus (4), Copilot (3)
Total	**97**	**10 unique agents**

supporting professional workflows across multiple domains. Since startup agents are typically organized around user needs and workflow capabilities rather than predefined disciplinary boundaries, a single agent may contribute tasks to multiple domains. We therefore assign domain labels at the task level based on the professional context and objectives of each task.

Table 7 summarizes the selected agents and their contributions across domains. The task distribution reflects the workflows identified through our sourcing and user interview process, rather than a predefined allocation of agents to domains. The complete task-level mapping between benchmark instances and their source agents is provided in our anonymous code repository.

E RUBRICS OF STARTUPBENCH

Rubric Dimensions. To facilitate consistent evaluation across heterogeneous real-world tasks, we organize all evaluation rubrics into six high-level categories according to the primary capability they assess. Rather than being tied to specific application domains, these categories capture common dimensions of deliverable quality that recur across office tasks, including correctness, completeness, data processing, professional reasoning, engineering quality, and presentation. This taxonomy enables more interpretable analysis of model strengths and weaknesses while maintaining a unified evaluation framework across different task types.

- **Structure & Completeness.** Evaluates whether the required deliverables are fully produced with the correct organization, file structure, required components, and overall completeness.
- **Calculation Precision.** Evaluates the correctness of numerical results, logical reasoning, formula execution, factual consistency, and other objective computations.
- **Information Integration.** Evaluates data cleaning, transformation, aggregation, statistical analysis, feature engineering, modeling, and other structured data manipulation workflows.
- **Domain-Specific Compliance.** Evaluates whether outputs satisfy domain-specific professional requirements, including business logic, financial principles, legal reasoning, medical practice, educational standards, and other expert knowledge.
- **Engineering & Format.** Evaluates implementation quality, coding conventions, spreadsheet engineering, document formatting, reproducibility, robustness, naming conventions, and compliance with technical specifications.
- **Output & Presentation.** Evaluates the quality of visual presentation and communication, including charts, layouts, formatting, readability, report organization, and overall usability of the final deliverables.

Table 8 summarizes the distribution of all 2,453 rubrics in StartupBench. Calculation Precision constitutes the largest rubric category (38.9%), reflecting the importance of producing objectively correct work products. Structure & Completeness and Domain-Specific Compliance together account

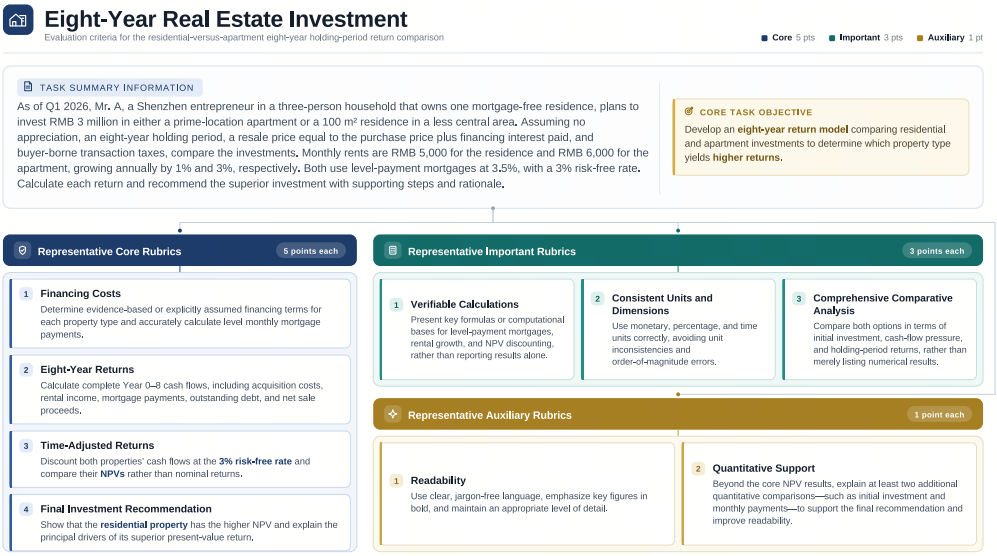


Figure 3: A representative example of a rubric of a task spanning 3 levels. Core rubrics assess the essential financial reasoning and calculations required to solve the investment problem, Important rubrics evaluate the correctness and completeness of the supporting analysis, while Auxiliary rubrics capture presentation quality and additional quantitative support.

for another 42.8%, highlighting that successful completion of realistic office workflows requires not only correct computations but also complete deliverables and professional domain reasoning.

Table 8: Distribution of rubric categories in StartupBench.

Category	Percentage
Structure & Completeness	27.31%
Calculation Precision	38.89%
Information Integration	9.46%
Domain-Specific Compliance	15.49%
Engineering & Format	6.20%
Output & Presentation	2.65%
Total	**100.00%**

Rubric Importance Levels. To account for differences in the importance of individual requirements to successful task completion, we assign each rubric to one of three importance levels. **Core** rubrics capture requirements that directly determine whether the primary user needs are satisfied; violating them would substantially compromise the correctness, reliability, safety, or usability of the final deliverable. **Important** rubrics represent key requirements for high-quality task completion whose satisfaction materially affects the overall quality of the deliverable, but whose individual violation does not necessarily render it unusable. **Auxiliary** rubrics capture lower-level requirements whose omission primarily affects details, user experience, visual appeal, or polish rather than the fundamental usability of the deliverable. We assign weights of **5**, **3**, and **1** to Core, Important, and Auxiliary rubrics, respectively, and use these weights when aggregating rubric-level judgments into the task score. A typical case of task containing rubrics of different importance levels is shown at Figure 3.

Across all tasks, Core, Important, and Auxiliary rubrics account for an average of 59.10%, 34.33%, and 6.57% of the total task weight, respectively, ensuring that task scores are primarily determined by requirements that materially affect successful task completion.

Table 9: Human–judge agreement on representative models, covering all 97 StartupBench tasks.

	Rubric-level (%)				Task-level (%)
Model	**Accuracy**	**Precision**	**Recall**	**F1**	**Success-Alignment**
Seed-2.1-Pro	92.46	97.63	90.59	93.98	94.85
GPT-5.5	91.36	96.80	91.18	93.91	91.75
Overall	**91.91**	**97.19**	**90.90**	**93.94**	**93.30**



Figure 4: Performance across rubric dimensions. Importance-weighted pass rates (%) of different models on the six rubric dimensions in StartupBench. Higher values indicate better performance under the corresponding evaluation dimension.

F Human–judge Agreement of Proposed Evaluation Framework in StartupBench

The detailed result of Human–judge agreement on GPT-5.5 and Seed-2.1-Pro is shown at Table 9. Rubric-level metrics evaluate binary rubric satisfaction, while task-level agreement compares success decisions under the threshold of 90.

G Model Performance across Different Rubric Dimensions

The detailed result of model performance across different rubric dimensions in StartupBench is shown at Figure 4.

H Output-format Compliance

The performance of models on output-format compliance requirements is shown at Table 10.

Table 10: Output-format compliance on the 56 tasks with explicit output-file rubrics, ranked by compliance rate.

Model	Compliance (%)
Kimi-K3	97.6
Seed-2.1-Pro	95.2
GPT-5.6-sol	94.6
Qwen-3.6-Max	94.6
DeepSeek-V4-Pro	94.0
GPT-5.5	93.5
Kimi-K2.6	92.9
Gemini-3.1-Pro	86.9
GLM-5.1	86.3

Table 11: Task-normalized rubric pass rate (%).

Model	Auxiliary	Important	Core
Kimi-K3	77.89	75.12	73.40
GPT-5.6-sol	77.55	74.47	73.07
GPT-5.5	75.12	75.09	72.07
Seed-2.1-Pro	71.08	70.16	65.89
GLM-5.1	64.90	61.75	60.83
Kimi-K2.6	66.64	63.56	57.82
Qwen-3.6-Max	65.39	62.17	58.08
DeepSeek-v4-Pro	66.50	62.97	60.05
Gemini-3.1-Pro	53.49	48.41	50.13
Average	**68.73**	**65.97**	**63.48**

I Model Performance across Different Rubric Importance Levels

The detailed performance of models on different rubric importance levels is shown at Table 11. The pass rate on different levels are calculated by the following metric: for each model–task–trial, task-level pass rates are computed within each importance level, and then averaged over all trials.

J Failure on Core Requirements

As shown in Figure 5, in a mechanical process task, models complete peripheral workbook requirements but fail on the core technical operations: extracting dimensions accurately from engineering drawings, validating specifications against the provided documents, and linking identified discrepancies to the corresponding rectification actions. These failures prevent the resulting workbook from fulfilling its primary purpose even when many auxiliary requirements are successfully completed.

K Examples of Behavior-level Failure Modes

K.1 Complex Instruction Following

The task requires an HR payroll workbook built from multiple source sheets under structural, computational, and formatting constraints. Although the model produces all required worksheets, formulas, charts, and dashboard layouts, artifact-level evaluation finds missing cached values needed for downstream verification. The polished but unverifiable workbook exemplifies partial compliance with visible requirements while less obvious, equally essential constraints remain unmet (Figure 6).

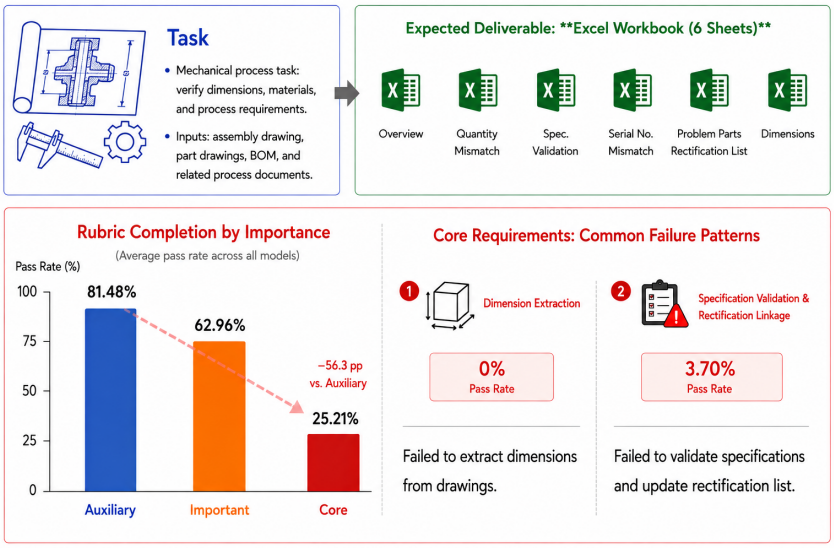


Figure 5: A representative case of models failing on core rubrics though passing auxiliary rubrics on StartupBench.

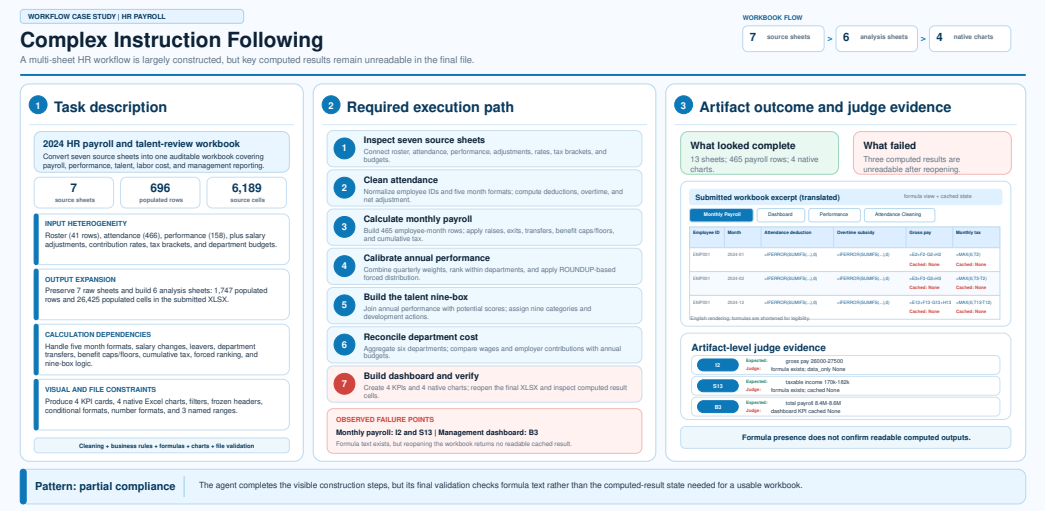


Figure 6: A representative case of complex instruction following failure. Although the generated workbook appears complete, artifact-level evaluation reveals that several critical acceptance conditions remain unsatisfied, illustrating a common pattern of partial compliance.

K.2 Self-Verification Hallucination

A representative example is shown in Fig. 7. The task requires the model to filter all level-11 members, simulate route check-ins until each member reaches the level-12 threshold, and generate an Excel workbook satisfying a set of precise value and formatting constraints. The produced workbook appears largely complete, containing the required worksheet, member records, and output columns, leading the model to conclude that the task has been successfully finished. However, artifact-level inspection reveals multiple acceptance-critical errors, including incorrect effective-point values and mismatched member names. The underlying issue is not that the model fails to perform the required computation, but that it never verifies the generated artifact itself. Instead, it treats a successful execution summary as evidence that the final deliverable has been validated. Consequently, localized yet user-critical errors remain unnoticed, illustrating a form of *self-verification hallucination*,

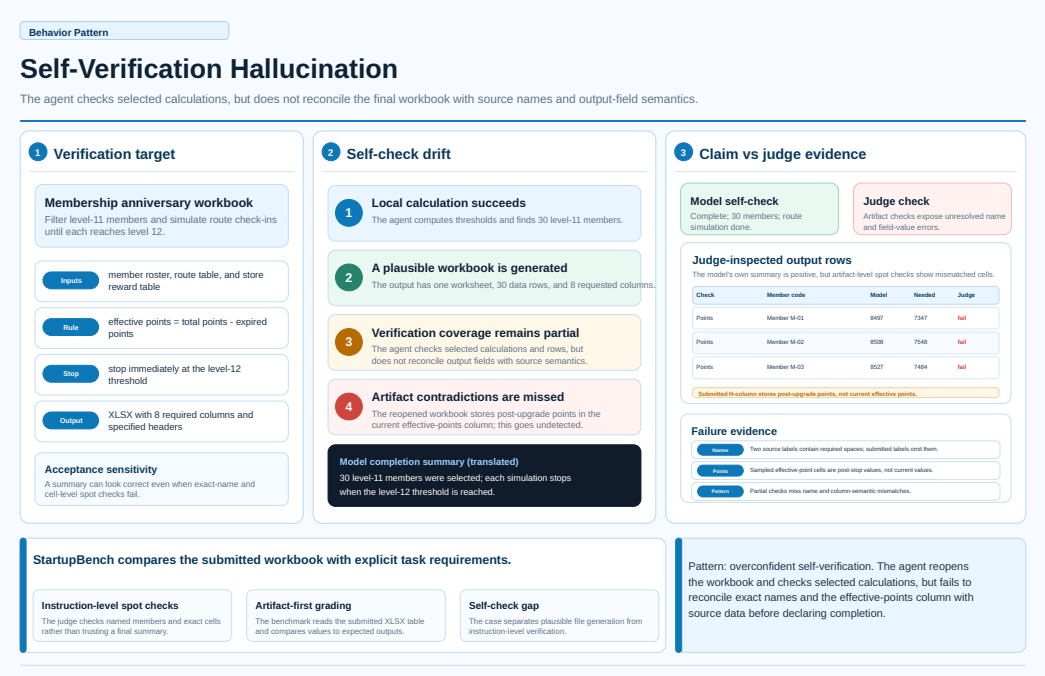


Figure 7: A representative example of self-verification hallucination. The model generates a plausible workbook and concludes that the task has been completed based on its execution process. However, artifact-level spot checks expose acceptance-critical errors in the final deliverable, showing that the model mistakes a progress summary for actual verification instead of validating the generated artifact against the original task requirements.

where confidence is derived from the reasoning process rather than from independently checking the completed artifact against the original task requirements.

K.3 Domain-Specific Failure

A representative example of failure caused by insufficient domain-specific clinical actionability is shown at Figure 8. The model generates a well-formatted multidisciplinary treatment plan that appears clinically professional, yet violates multiple safety-critical treatment constraints. The failure stems not from poor writing quality, but from an inability to convert medical knowledge into actionable and clinically reliable decisions.

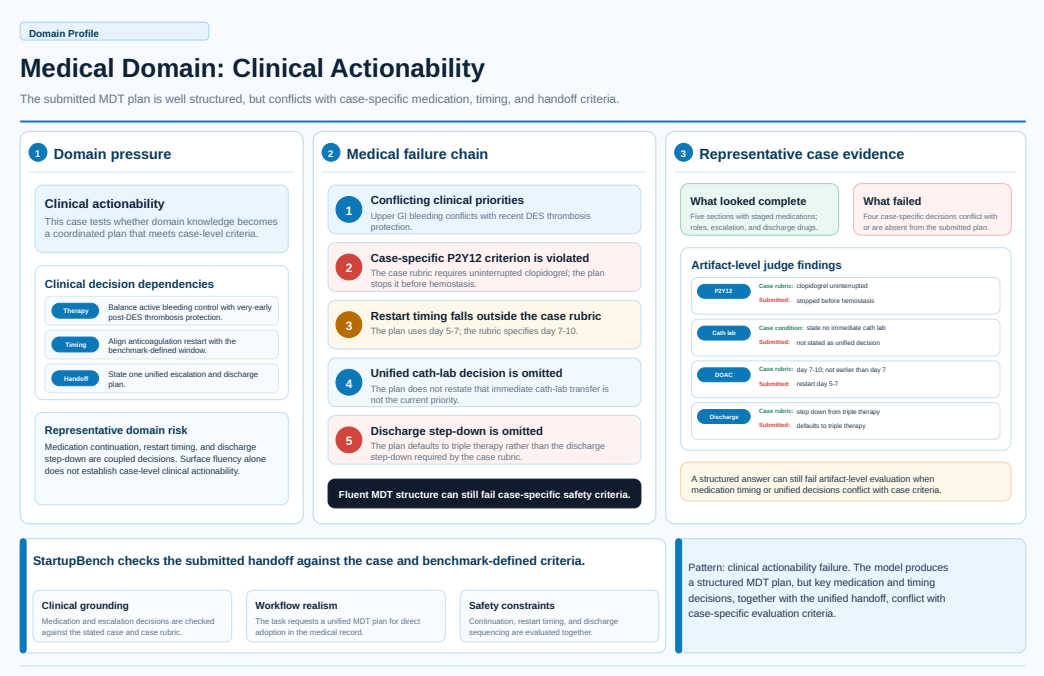


Figure 8: A representative example of failure caused by insufficient domain-specific clinical actionability.