PROFESSIONALBENCH: BENCHMARKING AI AGENTS ON REAL-WORLD PROFESSIONAL WORKFLOWS

Anonymous authors
Paper under double-blind review

ABSTRACT

As AI agents handle files, use tools, and produce complex deliverables, evaluating their ability to perform professional work becomes increasingly important. Recent benchmarks increasingly adopt workflow-based tasks to improve task realism. However, workflows may undergo *Workflow Compression* during task construction: responsibilities that belong to the designated role are omitted or resolved in advance, preventing direct evaluation of whether agents can fulfill them and where they fall short. To capture real professional workflows, we introduce **ProfessionalBench**, spanning 30 subdomains across finance, law, and healthcare. Experts compare tasks against source workflows and use a *Know–Act–Close (KAC)* formulation to preserve interdependent responsibilities for professional judgment, execution, and closure. Expert-reviewed *Dense Rubrics*, aligned with KAC responsibilities, support hybrid verification of requirement completion and deliverability, while trajectory analysis examines the failure mechanisms behind unmet requirements. We evaluate seven frontier agents on ProfessionalBench. Results reveal that even the strongest agents achieve pass rates below 40%. Workflow Compression interventions show thatomitting or pre-resolving responsibilities increases mean scores, with larger gains from compressing Act (+4.1 points) and Close (+3.7) than Know (+2.2). Trajectory-based diagnosis finds failures across all three responsibilities, with the largest shares attributed to Act in finance, Know in law, and Close in healthcare. Together, these findings show that preserving role-specific responsibilities exposes substantial professional headroom and identifies where agents need to improve.

1 INTRODUCTION

Recent advances in large language model (LLM) agents have substantially improved their capabilities in long-horizon reasoning, information retrieval, code execution, and tool use, expanding their functional scope from relatively constrained question answering and isolated operations toward increasingly complex real-world work. Evaluating such agents therefore requires moving beyond knowledge acquisition, reasoning accuracy, and individual actions to address a more consequential question: *can current agents reliably perform professional work in realistic occupational settings?* To address this question, benchmarks for professional agents have increasingly incorporated authentic workplace materials, open-ended deliverables, enterprise software, and domain-specific tools, often with domain experts involved in designing both tasks and evaluation criteria. This shift has moved agent evaluation from isolated capability testing toward end-to-end workflows with explicit business objectives, professional constraints, and delivery requirements.

Despite rapid progress in professional-agent benchmarking, real-world provenance alone does not guarantee a faithful and comprehensive representation of professional work. Expanding across domains and increasing the number of task instances do not necessarily provide systematic coverage of the distinct professional units and canonical workflows within each domain. More importantly, even when a workflow is included in a benchmark, the instantiated task may fail to preserve the critical responsibilities that a professional would normally be expected to undertake autonomously. During task construction, responsibilities such as evidence assessment, execution planning, or outcome verification may be omitted altogether, or effectively resolved in advance through pre-specified case conclusions, complete execution procedures, or verification results. The agent is then required to complete only the residual portion of the original workflow, without independently addressing the

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.

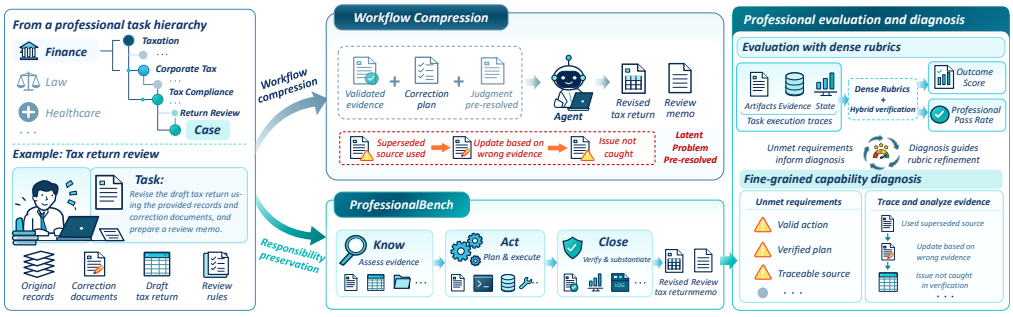


Figure 1: **Position of ProfessionalBench.** We organize tasks hierarchically, preserve Know–Act–Close responsibilities to address Workflow Compression, and enable fine-grained evaluation.

corresponding professional problem or the dependencies it entails. We refer to this loss of critical professional responsibilities during task instantiation as *Workflow Compression*. Workflow Compression changes not only the task itself, but also the capabilities that the benchmark can actually measure: once a professional responsibility has already been resolved during task construction, the resulting task no longer directly evaluates whether the agent can autonomously fulfill that responsibility. Consequently, finer-grained evaluation of the remaining task cannot compensate for the absence of a direct test of the agent's ability to fulfill that responsibility.

Research Question

How can we systematically collect real-world professional workflows and transform them into evaluation tasks that preserve critical responsibilities and their dependencies, thereby enabling a more comprehensive and fine-grained characterization of the boundaries of agents' professional capabilities?

To address this question, we introduce **ProfessionalBench**, a benchmark of real-world professional workflows spanning finance, law, and healthcare. Drawing on occupational resources, industry standards, and expert knowledge, we organize task sourcing and construction according to a *Domain–Subdomain–Professional Unit–Anchor Workflow–Case* hierarchy, and use identified coverage gaps to guide further expansion. At the task level, we characterize professional depth by the extent to which critical responsibilities and their dependencies are preserved, and operationalize this principle through Know–Act–Close (KAC), requiring agents to independently gather evidence and form professional judgments, plan and execute accordingly while advancing the relevant business state, and verify outcomes, handle in-scope exceptions, and determine whether the work is complete. Experts compare each task against its source workflow to identify and restore responsibilities that would otherwise be omitted or resolved in advance, while preserving its business objective, scope of responsibility, and professional completion criteria.

To assess whether these responsibilities are fulfilled, we construct expert-reviewed *Dense Rubrics* that link KAC duties and functional capabilities to both outcome conditions and traceable evidence of professional judgment, execution, and verification. Programmatic and agent-based verifiers inspect deliverables, evidence records, and final business states. *Outcome Score* summarizes satisfaction of responsibility-aligned criteria; *Professional Pass* requires meeting a task-specific threshold and critical criteria. We evaluate seven frontier model–harness configurations and conduct paired Workflow Compression interventions. Analysis show that Act-compressed variants exhibit the largest performance variation, while unmet criteria concentrate in Act and Close, which suggests that agents can complete parts of a task yet fail the execution and closure responsibilities.

Our contributions are threefold:

- **Responsibility-Preserving Benchmark.** We introduce **ProfessionalBench**, which systematically preserves critical responsibilities and their dependencies in real-world professional work through structured within-domain task organization and expert-guided Know–Act–Close construction.
- **High-Resolution Evaluation.** We combine expert-reviewed Dense Rubrics with hybrid verification to assess requirement completion and professional deliverability. Rubric-level localization and trajectory analysis further characterize unmet requirements and failure mechanisms.
- **Professional Capability Analysis.** We systematically evaluate frontier agent frameworks and use *Workflow Compression* interventions to reveal that responsibility preservation affects measured performance, capability profiles, and the observability of professional bottlenecks largely.

2 Related Work

Professional Task Evaluation. Recent benchmarks have shifted toward evaluating AI systems on occupationally relevant and open-ended tasks. GDPval (Patwardhan et al., 2025) studies economically valuable work from industry professionals, while APEX (Vidgen et al., 2025), PRBench (Akyürek et al., 2026), and XpertBench (Liu et al., 2026) evaluate professional tasks with expert-defined criteria. ALE (Sun et al., 2026) further introduces occupational taxonomy and workflow-family organization for systematic task construction. Other efforts explore specific settings, including workplace delegation (Li et al., 2026b), financial agents (Zeng et al., 2025), and coding-oriented professional workflows (Team et al., 2026). However, existing benchmarks largely rely on task sampling and lack systematic coverage of professional units and workflow responsibilities. ProfessionalBench addresses this gap by constructing an expert-calibrated hierarchy across three domains to guide workflow selection and coverage expansion.

Long-Horizon Workflow Evaluation. Interactive benchmarks evaluate whether agents can execute multi-step workflows under changing environments. WebArena (Zhou et al., 2024) and OSWorld (Xie et al., 2024) introduce realistic browser and desktop environments, while WorkArena (Drouin et al., 2024), AppWorld (Trivedi et al., 2024), and TheAgentCompany (Xu et al., 2025) extend evaluation to enterprise applications and workplace interactions. Recent benchmarks further study complex tool use, such as Toolathlon (Li et al., 2026a), SaaS-Bench (Shi et al., 2026), and Workflow-GYM (Zhu et al., 2026). Nevertheless, longer trajectories do not guarantee faithful professional workflows. ProfessionalBench introduces Workflow Compression analysis and expert-guided Know–Act–Close construction to identify missing responsibilities and dependencies.

Rubric-Based Evaluation and Diagnosis. LLM-based evaluation methods, including LLM-as-a-judge (Zheng et al., 2023) and G-Eval (Liu et al., 2023), enable scalable assessment of open-ended outputs. Professional benchmarks further incorporate expert-designed criteria, such as PRBench (Akyürek et al., 2026), XpertBench (Liu et al., 2026), HealthBench (Arora et al., 2025), and LexRubric (Chen et al., 2026). Beyond final-answer scoring, recent work analyzes intermediate traces and criterion-level failures, including JobBench (Li et al., 2026b), BigFinanceBench (Wang et al., 2026), and ALE (Sun et al., 2026). However, existing frameworks provide limited diagnosis of unmet requirements and failure causes. ProfessionalBench combines responsibility-aligned dense rubrics with hybrid verification for deliverable assessment and fine-grained failure analysis.

3 Overview of ProfessionalBench

We begin by outlining our design principles and choices. We then describe ProfessionalBench's construction, quality-control measures, and some descriptive statistics.

3.1 Design Principles

ProfessionalBench is designed to preserve what agents must independently accomplish in professional work and support fine-grained capability evaluation. Its design follows three principles:

- **Professional Responsibility Fidelity.** Tasks must preserve source workflows' critical responsibilities and the dependencies among professional judgment, execution, and closure, rather than omitting or pre-resolving responsibilities that agents should fulfill autonomously.
- **High-Resolution Verifiability.** Professional responsibilities must be evaluated against explicit acceptance criteria using observable evidence from artifacts and business states. Evaluation should localize unmet requirements across responsibilities and capabilities while distinguishing completion from professional deliverability and actionable capability gaps.
- **Structured Professional Coverage.** Professional work is organized around recurring responsibilities and workflows rather than isolated prompts. ProfessionalBench constructs an expert-calibrated task space and selects representative Anchor Workflows to achieve systematic coverage across domains and Professional Units with balanced workflow.

Guided by these principles, we formalize responsibility-preserving tasks (Sec. 3.2), describe their construction and evaluation (Secs. 3.3–3.5), and summarize the resulting benchmark (Sec. 3.4).

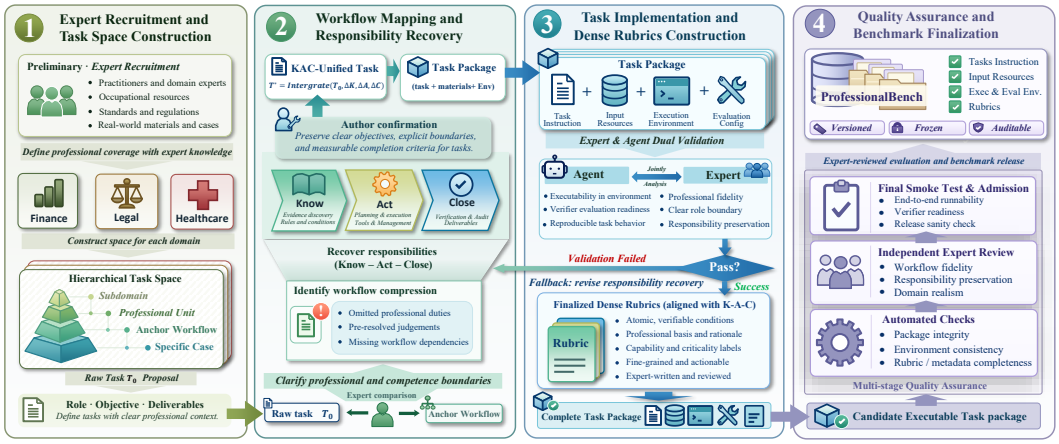


Figure 2: **Construction pipeline of ProfessionalBench.** The pipeline transforms authentic professional workflows into responsibility-preserving, executable, and verifiable KAC-Unified Tasks through expert-guided construction and quality assurance with workflow fidelity preservation.

3.2 Responsibility-Preserving Workflow Formulation

To operationalize Professional Responsibility Fidelity, ProfessionalBench adopts the *Anchor Workflow* as the basic unit of task formulation: a self-contained unit of professional work with a designated role, business objective, and completion criteria. Relative to the designated role and task boundary, *Workflow Compression* occurs when task construction omits or pre-resolves critical responsibilities that agents should otherwise fulfill. We characterize *Professional Depth* by the extent to which a task preserves the critical responsibilities and dependencies of its source workflow.

Know–Act–Close Responsibility Criteria. To translate this notion into task-design criteria, domain experts identify the responsibilities of each Anchor Workflow across three coupled stages:

- *Know: Form reliable judgments from evidence.* The agent should independently discover and assess evidence, forming evidence-grounded judgments rather than relying on pre-resolved hints.
- *Act: Advance professional execution.* The agent should apply its judgments through execution, adapting to intermediate results and exceptions rather than following pre-specified procedures.
- *Close: Verify and substantiate completion.* The agent should verify outcomes against completion criteria, addressing in-scope discrepancies and substantiating completion with traceable evidence.

Know, Act, and Close are not independent subtasks or a prescribed one-pass sequence. Judgments formed during Know must guide Act, while Close verifies the resulting outcomes and may trigger renewed evidence assessment or corrective action. A responsibility-preserving task must retain these dependencies around a shared professional objective.

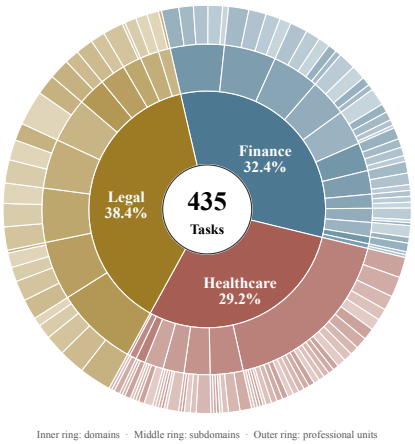
3.3 Expert Construction and Quality Assurance

As shown in Figs.2, ProfessionalBench employs a four-stage, expert-led, model-assisted pipeline to transform authentic professional tasks into responsibility-preserving and professional tasks.

Stage 1: Task-Space Planning and Expert Sourcing. ProfessionalBench first constructs a hierarchical professional task space spanning finance, law, and healthcare. Drawing on occupational resources (e.g., O*NET activities), industry standards, and expert input, we organize professional work into *30 subdomains* and *127 Professional Units*. Domain experts define representative *Anchor Workflows* for each unit, specifying responsible roles, business objectives, and independently verifiable completion criteria. Based on this task space, practitioners contribute raw tasks T_0 together with supporting professional materials, including relevant records, standards, regulations, and practice guidelines, which provide the basis for subsequent responsibility analysis.

Stage 2: Responsibility Recovery and Task Instantiation. Each raw task T_0 is reviewed against its source Anchor Workflow W by domain experts who have not participated in task creation. Experts identify omitted or pre-resolved responsibilities as KAC gaps $(\Delta_K, \Delta_A, \Delta_C)$, and propose recovery

Table 1: Domain-wise composition and coverage density of ProfessionalBench. The sunburst (left) shows the domain-specific taxonomy; all taxonomy counts are computed within domains, and similarly named nodes in different domains are treated as distinct categories.



Statistic	Healthcare	Finance	Legal	Total
Benchmark scale				
Subdomains	8	12	10	30
Professional Units	58	41	28	127
Anchor Workflows	113	108	134	355
Selected Tasks	127	141	167	435
Task Share	29.2%	32.4%	38.4%	100.0%
Coverage density				
Tasks per Subdomain	15.9	11.8	16.7	14.5
Tasks per Professional Unit	2.2	3.4	6.0	3.4
Tasks per Anchor Workflow	1.12	1.31	1.25	1.23
Professional Units per Subdomain	7.3	3.4	2.8	4.2
Anchor Workflows per Professional Unit	1.9	2.6	4.8	2.8

operations with their professional basis and dependency relationships. After review and confirmation, these modifications are integrated into the final task $T^* = \text{Integrate}(T_0, \Delta_K, \Delta_A, \Delta_C)$. The resulting T^* is defined as a *KAC-Unified Task*, preserving the original workflow objective, responsibility boundary, and completion criteria while restoring critical professional responsibilities. Each task is then packaged with standardized metadata and execution materials for subsequent environment construction and evaluation.

Stage 3: Task Implementation and Dense Rubrics Construction. KAC-Unified Tasks are instantiated as candidate task packages containing task instructions, input resources, a Docker execution environment, and evaluation configurations. Domain experts translate retained responsibilities into Dense Rubrics with atomic acceptance conditions, expert-assigned weights, and observable verification targets, including required work records and final deliverables. Agent-assisted validation checks executability and evaluation readiness, while experts assess professional fidelity, role boundaries, and responsibility preservation. Task-design failures are returned to Stage 2 for revision.

Stage 4: Quality Assurance and Benchmark Finalization. Candidate packages undergo automated integrity and consistency checks, independent expert review, and final end-to-end smoke tests. These checks assess package completeness, workflow fidelity, rubric alignment, and verifier readiness. Packages failing a check are returned for revision and rechecked before admission. Accepted tasks, environments, and evaluation configurations are versioned and frozen for benchmark release.

3.4 Benchmark Composition

As summarized in Table 1, ProfessionalBench comprises **435** validated KAC-Unified tasks: **141** in finance, **167** in law, and **127** in healthcare, covering **30** domain-specific subdomains, **127** professional units, and **355** anchor workflows across diverse professional settings. These counts exclude workflow compression intervention variants. The complete taxonomy, including the professional units and anchor workflows covered within each domain, is provided in Appendix F.

3.5 Evaluation Protocol

ProfessionalBench evaluates each run through evidence-grounded hybrid verification and reports two metrics: an *Outcome Score* for requirement completion and a binary *Professional Pass* for professional acceptability across diverse real-world professional workflows.

Evidence-Grounded Hybrid Verification. After each run, we freeze the agent's workspace and environment state. Each Rubric is evaluated against its designated target in the task-required records, including final artifacts, intermediate work products, and responsibility-related evidence. *Programmatic Verifiers* check deterministic conditions, while an *Agent Verifier* inspects the frozen evidence

Table 2: **Main results on ProfessionalBench** We evaluate seven frontier models on 90 professional tasks, with 30 tasks from each domain. Score and Pass denote Outcome Score and Pass Rate.

Model	Finance				Law				Healthcare			
	Score (0–100)	Pass (%)	Delay (min)	Cost ($)	Score (0–100)	Pass (%)	Delay (min)	Cost ($)	Score (0–100)	Pass (%)	Delay (min)	Cost ($)
GPT-5.6 Sol	60.63	**36.7**	**5.38**	1.78	**63.11**	**30.0**	13.32	4.46	**66.98**	33.3	**6.58**	1.76
Claude Opus 5	**63.72**	33.3	14.78	2.68	60.74	26.7	31.35	4.49	64.37	**36.7**	15.95	2.86
Gemini 3.8 Flash	41.91	13.3	9.78	0.81	46.53	10.0	12.23	0.84	51.14	16.7	13.80	0.71
DeepSeek-V4.1-Flash	52.04	6.7	15.22	**0.25**	52.04	13.3	30.93	0.79	55.05	23.3	11.10	**0.26**
GLM-5.2	50.16	10.0	13.10	0.49	45.14	6.7	**10.67**	**0.49**	49.71	20.0	9.88	0.40
Qwen3.8-Max	53.88	26.7	51.40	2.64	42.47	6.7	63.33	2.11	58.82	23.3	27.53	1.43
Kimi K3	57.82	20.0	14.90	1.43	53.34	20.0	38.68	6.37	57.51	16.7	12.58	0.98

to assess open-ended professional requirements against expert-approved criteria. Each verdict is grounded in supporting evidence and accompanied by a concise rationale.

Outcome Scoring and Professional Gates. For task t, let $\mathcal{R}_t$ denote ordinary scored rubric items, $w_{t,i}$ their signed weights, and $h_{t,i} \in \{0,1\}$ whether the corresponding predicate is true. Positive-weight items describe desirable properties, whereas negative-weight items describe failure events. Let $\mathcal{H}_t$ denote disqualifying hurdle predicates and $\mathcal{C}_t$ denote critical requirements that must be satisfied. With $g_t = \prod_{j\in\mathcal{H}_t}(1-h_{t,j})$, we define

$$S_t = g_t \max\left\{0, \frac{\sum_{i\in\mathcal{R}_t} w_{t,i}h_{t,i}}{\sum_{i\in\mathcal{R}_t}\max(w_{t,i},0)}\right\}, \quad \text{Pass}_t = \mathbf{1}\{S_t \geq \tau_t\}\prod_{i\in\mathcal{C}_t} h_{t,i}. \tag{1}$$

Thus, penalties reduce the normalized score, any triggered hurdle sets it to zero, and Professional Pass additionally requires all critical requirements to be satisfied. We set $\tau_t = 0.85$ and report Outcome Score as $100S_t$. All weights and gates are fixed before evaluation.

4 Experiments

4.1 Experimental Setup

Benchmark. We evaluate a stratified subset of 90 tasks from ProfessionalBench, with 30 tasks each in finance, legal, and healthcare. The subset is sampled from the frozen 435-task benchmark; its exact subdomain and Professional Unit composition is reported from the task manifest. The tasks cover evidence assessment, business execution, and professional verification with diverse outputs, including reports, spreadsheets, and structured records.

Models. We evaluate seven frontier models: GPT-5.6 Sol, Claude Opus 5, Gemini 3.8 Flash, DeepSeek-V4.1-Flash, GLM-5.2, Qwen3.8-Max, and Kimi K3. All models except DeepSeek-V4.1-Flash are run through Harbor; DeepSeek-V4.1-Flash uses an independently configured harness–model pair. Configuration details, including model snapshots, harness versions, tool permissions, and execution environments, are provided in Appendix C.

Evaluation. After each run, we freeze the workspace and environment state and provide the evaluator with rubrics, reference results, and verification scripts. Programmatic Verifiers check deterministic conditions, while GPT-5.6 Sol with the Codex harness serves as the Agent Verifier for open-ended professional requirements. Verdicts are grounded in the submitted artifacts, execution records, and final business states. We report Outcome Score and Professional Pass following Sec. 3.5.

4.2 Main Results

Table 2 reports performance on ProfessionalBench across finance, law, and healthcare, and Figure 3 summarizes the performance-efficiency trade-offs.

Frontier agents achieve strong completion but limited acceptance. GPT-5.6 Sol and Claude Opus 5 consistently achieve the strongest performance across domains, forming the Pareto frontier

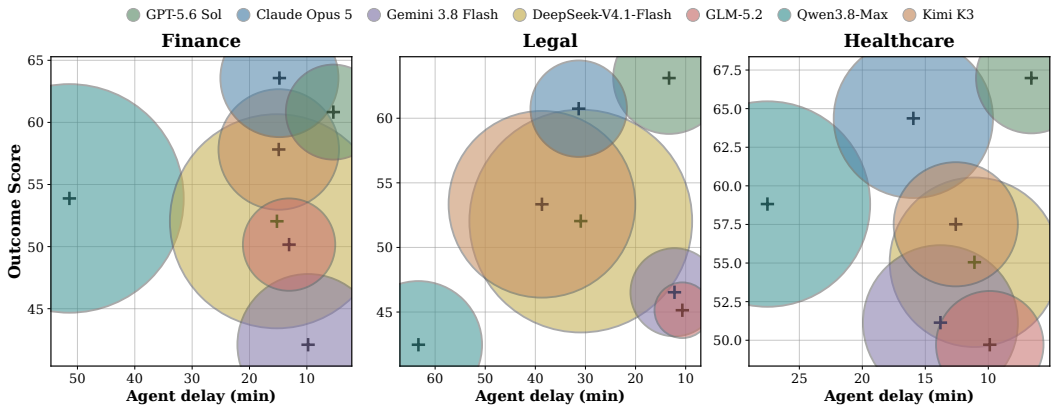


Figure 3: **Model performance and efficiency.** Comparison of model performance, cost, and delay across domains. Bubble size indicates token consumption.

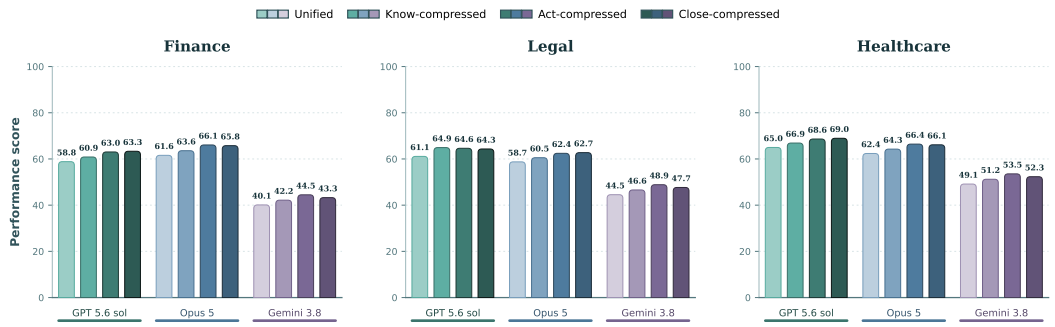


Figure 4: **Workflow compression variants.** Performance scores for Unified, Know-compressed, Act-compressed, and Close-compressed are shown for three model in finance, legal, and healthcare.

in Figure 3. However, even these leading configurations achieve Professional Pass Rates below 40%, revealing a gap between requirement completion and end-to-end professional acceptance.

Cost–Performance Trade-off. Token consumption and monetary cost exhibit different trade-offs across models. DeepSeek-V4.1-Flash achieves the lowest cost per task ($0.25–$0.79 across domains), while Figure 3 shows that DeepSeek-V4.1-Flash and Qwen 3.8-max consume substantially more tokens without a proportional improvement in Outcome Score. These results indicate that increased resource usage alone does not guarantee stronger professional workflow performance.

Delay-Performance Trade-off. Execution speed also varies across models. GPT-5.6 Sol achieves competitive performance with substantially lower delay than Claude Opus 5, while Qwen3.8-Max incurs longer execution times without corresponding performance gains. Overall, efficient execution requires balancing completion quality with computational cost and latency.

5 Discussion

5.1 Workflow Compression Interventions

We sample 10 tasks from ProfessionalBench and construct three paired variants for each: Know-compressed, Act-compressed, and Close-compressed. We evaluate each Unified task and its variants using three frontier agents to examine how responsibility compression affects measured performance, while keeping the business objective, role boundary, shared evidence, deliverable format, and execution budget fixed.

Variants' Results. Across matched task–model pairs, Workflow Compression consistently changes measured agent performance. Act compression produces the largest improvement over the full (Unified) task, increasing the mean Outcome Score by 4.08 %. Close compression yields the second-

Table 3: **Failure attribution taxonomy.** Failure categories used for trajectory-based diagnosis. Each observed failure is assigned to one primary category and one failure type based on evidence from the execution trajectory and final deliverables.

Primary category	Failure type	Core definition
Know	Domain Knowledge Gap	Missing or misapplied professional knowledge, domain rules, concepts, or judgment.
	Evidence Selection / Interpretation Error	Incorrect selection or interpretation of task evidence.
	Hallucination / Fabrication	Unsupported facts, data, citations, files, or claimed tool results.
Act	Wrong Strategy / Reasoning Error	An unsuitable method, analytical framework, calculation basis, or reasoning path.
	Implementation Bug	A local error in code, formulas, calculations, data processing, parameters, or state updates.
	Boundary / Safety Violation	Unauthorized actions, unsafe operations, or improper changes to protected files or state.
Close	Incomplete / Abandoned	Required steps, checks, content, or deliverables are omitted or left unfinished.
	Communication / Format Error	The final output is unclear or violates file, template, schema, precision, or delivery requirements.
	Non-actionable Output	Recommendations or plans are present but infeasible or insufficiently specific for execution.
Infrastructure	Timeout / Resource Limit	The run exceeds available wall-clock time, tokens, memory, or compute.
	Harness / Environment Error	The run fails because of tool, API, connection, environment, or agent-harness faults.

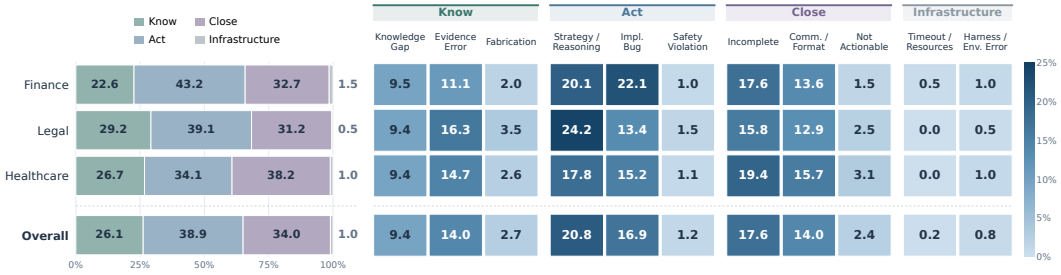


Figure 5: **Failure attribution across domains and capabilities.** Distribution of diagnosed failure modes across Know, Act, Close, and infrastructure categories for finance, legal, and healthcare tasks. Values represent the proportion of failures assigned to each category within each domain.

largest improvement of 3.68 %, followed by Know compression with an improvement of 2.18 %. This ordering generally holds across domains and models, although the magnitude of the effect varies. We additionally compare performance changes on rubric criteria shared between each compressed variant and its Unified counterpart, distinguishing genuine capability changes from differences introduced by modified evaluation scope.

Analysis. The ordering reflects the dependency structure of professional workflows. Act responsibilities require agents to translate judgments into executable plans, perform operations, and adapt to intermediate outcomes. When these responsibilities are pre-resolved, agents avoid execution errors and can satisfy downstream requirements that remain challenging in the Unified setting. Close compression produces the next-largest effect, as reducing verification and completion responsibilities allows agents to satisfy deliverable requirements without fully substantiating outcomes. In contrast, Know compression has a smaller impact, suggesting that evidence assessment and judgment formation remain important but are less frequently the primary bottleneck once downstream execution and closure requirements are simplified. These results demonstrate that Workflow Compression changes not only task difficulty, but also the capabilities that benchmarks can observe.

5.2 Failure-mode Analysis

We combine Dense Rubric verdicts with execution trajectories to distinguish unmet requirements from their underlying failure mechanisms. Rubrics locate unsatisfied acceptance conditions, while trajectory analysis identifies evidence-supported errors and omissions. Correlated rubric failures are grouped when they arise from the same upstream error. Figure 5 reports the composition of diagnosed failures rather than per-task failure probabilities.

Overall. Act and Close account for 38.9% and 34.0% of diagnosed failures, respectively, followed by Know (26.1%) and infrastructure (1.0%). The most frequent individual types are strategy or reasoning errors (20.8%), incomplete or abandoned work (17.6%), and implementation bugs (16.9%). These results highlight difficulties beyond evidence assessment and professional judgment: agents must also translate judgments into correct execution and carry required work through to completion.

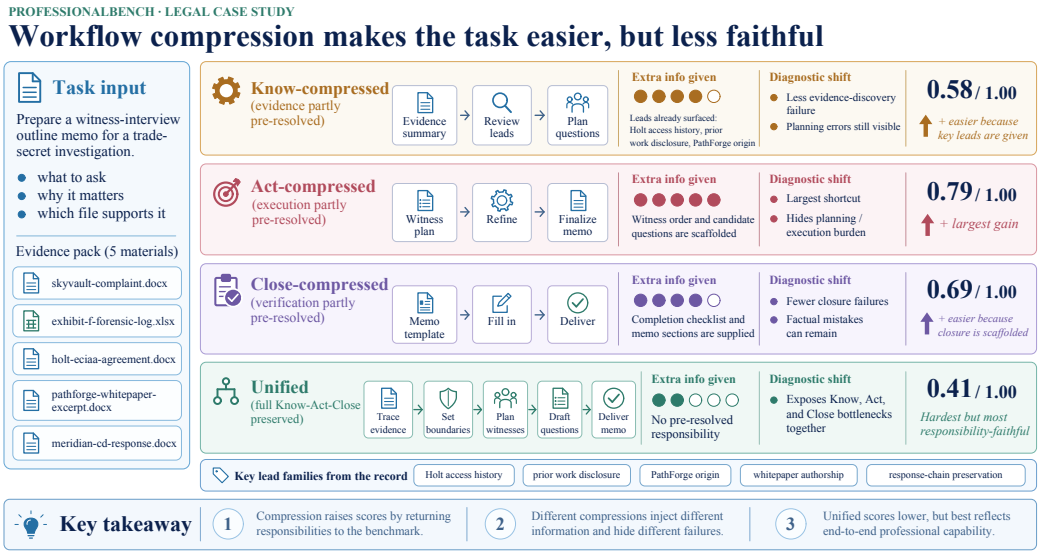


Figure 6: **Case Study.** A legal task from ProfessionalBench. The figure illustrates the task workflow, its Know–Act–Close responsibilities, and the evidence required to assess their completion.

Domain-Specific. Act is the largest failure category in finance (43.2%) and law (39.1%), but their leading failure types differ. Finance shows frequent implementation bugs (22.1%) and strategy or reasoning errors (20.1%), whereas law is led by strategy or reasoning errors (24.2%), followed by evidence selection or interpretation errors (16.3%). In healthcare, Close accounts for the largest share (38.2%), with incomplete or abandoned work (19.4%) and communication or format errors (15.7%) prominent within this category. These profiles distinguish domain-specific difficulties in execution, evidence handling, and delivery that aggregate performance scores alone do not reveal.

5.3 Representative Case Studies

The case requires a witness-interview outline memo for a trade-secret investigation. Given five materials, the agent must determine what to ask, why each question matters, and which file supports it. Unified preserves the responsibilities of evidence assessment, interview planning, and verification.

Compression Effects. Know-compressed provides key evidentiary leads; Act-compressed scaffolds witness ordering and candidate questions; Close-compressed supplies a memo template and completion checklist. In the illustrated comparison, scores increase from 0.41 under Unified to 0.58, 0.79, and 0.69, respectively. The largest increase occurs when interview planning and execution are partly pre-resolved, with the task objective and required deliverable unchanged.

Diagnostic Implications. Each intervention changes which failures remain observable: supplied leads reduce evidence-discovery demands, interview scaffolding bypasses planning challenges, and completion aids simplify closure while leaving factual mistakes possible. Higher scores need not indicate stronger end-to-end capability; they may reflect responsibilities resolved by the benchmark. Unified retains these demands, exposing limitations that compressed variants can conceal.

6 Conclusion

We introduced **ProfessionalBench**, a benchmark for evaluating AI agents on real-world professional workflows while preserving the responsibilities required for professional judgment, execution, and verification. By identifying *Workflow Compression* and operationalizing responsibility preservation through *Know–Act–Close*, ProfessionalBench enables more faithful evaluation and fine-grained diagnosis of agent capabilities. Experiments across finance, law, and healthcare show that strong task completion does not necessarily translate into professional acceptance, and that different responsibility omissions lead to distinct capability patterns. We hope ProfessionalBench provides a foundation for developing more reliable evaluations of AI agents in complex professional settings.

AI USE STATEMENT

In this work, we used generative AI tools to assist with literature search, literature summarization, and manuscript writing refinement. We also used AI tools during the preparation of synthetic evaluation data, with all generated content reviewed and verified by the authors. For the experiments, GPT-5.6 Sol with the Codex harness was evaluated as an Agent Verifier for open-ended criteria defined by domain experts; this usage is part of the evaluated system configuration rather than assistance in writing or conducting the research. The authors take full responsibility for the accuracy of the cited literature, experimental records, analyses, and final manuscript, including all AI-assisted content.

ETHICS STATEMENT

ProfessionalBench evaluates agent performance on tasks drawn from finance, law, and healthcare, where source materials and generated artifacts may involve confidential or sensitive information. Task materials should be distributed only when their use and release are authorized, with sensitive information removed or access restricted where appropriate. The benchmark is intended for research evaluation; its scores do not certify that an agent is suitable for unsupervised professional practice. In consequential settings, model outputs require review by appropriately qualified people before they inform financial, legal, or healthcare decisions.

REPRODUCIBILITY STATEMENT

The task hierarchy, Know–Act–Close formulation, construction process, scoring rules, and evaluation protocol are described in Sections 3 and 4. The supplementary materials provide the evaluated task manifest, task instructions and input resources, execution-environment specifications, rubric definitions, and verifier procedures. They also document model and harness versions, tool permissions, interaction budgets, and the aggregation procedure used to obtain the reported results. These materials identify the task and evaluation settings needed to reproduce the experiments, subject to any access restrictions on professional source materials.

REFERENCES

Afra Feyza Akyürek, Advait Gosai, Chen Bo Calvin Zhang, Vipul Gupta, Jaehwan Jeong, Anisha Gunjal, Tahseen Rabbani, Maria Mazzone, David Randolph IV, Mohammad Mahmoudi Meymand, Gurshaan Chattha, Paula Rodriguez, Diego A. Mares Buendia, Pavit Singh, Michael Liu, Subodh Chawla, Peter Cline, Lucy Ogaz, Ernesto Gabriel Hernández Montoya, Zihao Wang, Pavi Bhatter, Marcos Ayestaran, Bing Liu, and Yunzhong He. PRBench: Large-scale expert rubrics for evaluating high-stakes professional reasoning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42297–42325, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1958. URL https://aclanthology.org/2026.acl-long.1958/.

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

Yifan Chen, Haitao Li, Yiran Hu, Kaisong Song, Jun Lin, Yueyue Wu, Qingyao Ai, Min Zhang, and Yiqun Liu. Lexrubric: A rubric-guided diagnostic benchmark for open-ended legal tasks, 2026. URL https://arxiv.org/abs/2606.09389.

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.

Junlong Li, Wenshuo Zhao, Jian Zhao, Weihao Zeng, Haoze Wu, Xiaochen Wang, Rui Ge, Yuxuan Cao, Yuzhen Huang, Wei Liu, Junteng Liu, Zhaochen Su, Yiyang Guo, Fan Zhou, Lueyang Zhang, Juan Michelini, Xingyao Wang, Xiang Yue, Shuyan Zhou, Graham Neubig, and Junxian He. The tool decathlon: Benchmarking language agents for diverse, realistic, and long-horizon task execution, 2026a. URL https://arxiv.org/abs/2510.25726.

Yuetai Li, Yichen Feng, Zhangchen Xu, Zixian Ma, Kaiyuan Zheng, Fengqing Jiang, Xinghua Sun, Rulin Shao, Zichen Chen, Yue Huang, Xinyang Han, Brian Lee, Kayla Xu, Shenglai Zeng, Hang Hua, Xiangliang Zhang, Basel Alomair, Ranjay Krishna, Luke Zettlemoyer, Pang Wei Koh, Bhaskar Ramasubramanian, Luyao Niu, Xiang Yue, and Radha Poovendran. Jobbench: Aligning agent work with human will, 2026b. URL https://arxiv.org/abs/2605.26329.

Xue Liu, Xin Ma, Yuxin Ma, Yongchang Peng, Duo Wang, Zhoufutu Wen, Ge Zhang, Kaiyuan Zhang, Xinyu Chen, Yida Ding, Tianci He, Jiani Hou, Liang Hu, Ziyun Huang, Yongzhe Hui, Jianpeng Jiao, Chennan Ju, Yingru Kong, Yiran Li, Jiashuo Liu, Mengyun Liu, Luyao Ma, Fei Ni, Yiqing Ni, Pengbo Niu, Yueyan Qiu, Yanle Ren, Xinyu Shen, Zilin Shi, Zaiyuan Wang, Wenjie Yue, Chun Zhang, Shiyu Zhang, Xinyi Zhang, Kaiwen Zhao, Zhenwei Zhu, Shanshan Wu, Qi Zhao, and Wenhao Huang. Xpertbench: Expert level tasks with rubrics-based evaluation, 2026. URL https://arxiv.org/abs/2604.02368.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

Tejal Patwardhan, Rachel Dias, Elizabeth Proehl, Grace Kim, Michele Wang, Olivia Watkins, Simón Posada Fishman, Marwan Aljubeh, Phoebe Thacker, Laurance Fauconnet, Natalie S. Kim, Patrick Chao, Samuel Miserendino, Gildas Chabot, David Li, Michael Sharman, Alexandra Barr, Amelia Glaese, and Jerry Tworek. Gdpval: Evaluating ai model performance on real-world economically valuable tasks, 2025. URL https://arxiv.org/abs/2510.04374.

Kean Shi, Zihang Li, Tianyi Ma, Zengji Tu, Jialong Wu, Xinbo Xu, Qingyao Yang, Ruoyu Wu, Weichu Xie, Ming Wu, Jason Zeng, Michael Heinrich, Elvis Zhang, Liang Chen, Kuan Li, and Baobao Chang. Saas-bench: Can computer-use agents leverage real-world saas to solve professional workflows?, 2026. URL https://arxiv.org/abs/2605.15777.

Yiyou Sun, Xinyang Han, Weichen Zhang, Yuanbo Pang, Tianyu Wang, Yuhan Cao, Yixiao Huang, Chris Duroiu, Haoyun Zhang, Jeffrey Lin, Weishu Zhang, Tyler Zeng, Ying Yan, Bo Liu, Hanson Wen, Mingyang Xu, Xiaoyuan Liu, Zimeng Chen, Weiyan Shi, Amanda Dsouza, Vincent Sunn Chen, Patrick Bryant, Carl Boettiger, Yamini Rangan, Bradley Rothenberg, Kyle Steinfeld, Arvind Rao, Tapio Schneider, Georgios Yannakakis, Laure Zanna, Kaan Ozbay, Ida Sim, Tarek Zohdi, George Em Karniadakis, Jack Gallant, Teresa Head-Gordon, Yushan Li, Wenxi Deng, Tao Sun, Huiqi Wang, Zhun Wang, Justin Xu, Chris Yuhao Liu, Yafei Cheng, Rongwang Hu, Aras Bacho, Shengcao Cao, Zengyi Qin, Yixiong Chen, Hengduan Fan, Hao Liu, Lin Zeng, Shashank Muralidhar Bharadwaj, Litian Gong, Yingxuan Yang, Maojia Song, Ruheng Wang, Zongzheng Zhang, Honglin Bao, Shuo Lu, Jianhong Tu, Zhonghua Wang, Zheng Zhang, Zijiao Chen, Yanqiong Jiang, Zhendong Li, Bohan Lyu, Chang Ma, Peiran Xu, Benran Zhang, Shangding Gu, Haoyue Hua, Haoyang Li, Wanzhe Liao, Chengzhi Liu, Junbo Peng, Haoran Sun, Zechen Xu, Bo Chen, Jiayi Cheng, Yi Jiang, Keying Kuang, Yuan Li, Youbang Pan, Ziyan Rao, Alexander Schubert, Yifan Shen, Vincent Siu, Xiatao Sun, Kangqi Zhang, Xiaopan Zhang, Yuchen Zhu, Ishaan Singh Chandok, Lei Ding, Jingxuan Fan, Andrew Glover, Jiaming Hu, Yiran Hu, Wenbo Huang, Zixin Jiang, Haoran Jin, Lukas Kim, Ming Liu, Yang Liu, Alireza Rafiei, Xuhuan Shen, Kunyang Sun, Sophia Sun, Ting Sun, Eric Wang, Yixin Wang, Hanwen Xing, Sihan Xu, Yuzheng Xu, Zhongxing Xu, Zhiling Yan, Boqin Yuan, Ruiqi Zhang, Yifan Zhang, Zibo Zhao, Liana, Santanu Bosu Antu, Haoyue Bai, Carlo Bosio, Joseph Cavanagh, Patricia Cavazos-Rehg, Tianxing Chen, Xuewen Chen, Yipu Chen, Chenyu Zhu, Chen Dai, Stefano De Castro, Yunfu Deng, Kaustubh Dhole, Jiayuan Ding, Chenchen Du, Zhehang Du, Hao Fan, Run-Ze Fan, Hengyu Fu, Shi Gu, Yifan Gu, Charlie Guo, Baihe Huang, Baixiang Huang, Rimika Jaiswal, Zhihan Jiang, Ran Jin, Erin Kasson, Xin Lan, Joseph Lee, Deren Lei, Chenyu Li, Daofeng Li, Haitao

Li, Hongwei Li, Jingyan Li, Xiao Li, Yi Li, Yinsheng Li, Yuangang Li, Zhixu Li, Wenyu Liang, Longtai Liao, Kevin Qinghong Lin, Andy Zeyi Liu, Che Liu, Jiaming Liu, Kaiyuan Liu, Xuan Liu, Pan Lu, Wenbo Lv, Yicheng Lyu, Qiuyang Mang, Kyle Montgomery, Yuzhou Nie, Ruoxi Ning, Jorin Overwiening, Xu Pan, Layna Paraboschi, Core Francisco Park, Justin Purnomo, Swati Rajwal, Scott Rankin, Bixuan Ren, Yiren Rong, HaoYang Shang, Ventus Shaw, Fiona Shen, Jiawei Shen, Minqi Shi, Shi Qiu, Huaxiu Yao, Tianneng Shi, Jonah So, Vladislav Susoy, Hannah Szlyk, Haocheng Wang, Jialu Wang, Wei Wang, Xinyu Wang, Zehao Wang, Dowling Wong, Angela Wu, Dehao Wu, Fangyu Wu, Mengyuan "Millie" Wu, Yu Wu, Yuchen Wu, Yuhao Wu, Qingpo Wuwu, Weihang Xiao, Yongyi Xiong, Fan Xu, Ruiling Xu, Mingxuan Yan, Benjamin Yang, Jirong Yang, Sen Yang, Xiaoli Yang, Yushi Yang, Haoran Ye, Xiaohu Yu, Zhengming Yu, Chenlong Zhang, Chi Zhang, Hanning Zhang, Hanwen Zhang, Junge Zhang, Kunpeng Zhang, Song Zhang, Wenjin Zhang, Wenshuo Zhang, Ying Zhang, Yizhi Zhang, Brian Zhao, Qijian Zhao, Yimin Zhao, Yuhaohua Zheng, Liwei Zhou, Tianyue Zhou, Sichen Zhu, Siqi Zhu, Yan Zhu, Yishu Zhu, Jierui Zuo, Chonghao Cai, Helena Casademunt, Wenjia Chen, Cheng Cheng, Nawen Deng, Rao Fu, Tianfu Fu, Yifan Han, He Ren, Zhenyu He, Qiao Jin, Langlang Li, Yuetai Li, Sylvia Liu, Lu Lu, Luqing Zhou, Subhabrata Mukherjee, Yunqi Ouyang, Yin Ren, Dawei Shi, Haoran Wu, Zhiyue Wu, Hannah Yao, Zhuoran Yi, Jenny Yu, Rhea Zhan, Hang Zhou, Blake Zhu, Junfan Zhu, Alan Yuille, Yang Liu, Russell Alan Poldrack, Jiachen Li, Zhenglu Li, Molei Tao, Jing Huang, Wenqi Shi, Costas Spanos, Lichao Sun, Chenguang Wang, Orson Xu, Zhen Dong, Hector Gomez, Aylin Caliskan, Ali Emami, Haimin Hu, Zhi Li, Lihui Liu, Murphy Niu, Yi Shao, Jianxin Sun, Mikko Tolonen, Ting Wang, Sanjiv Das, Yanjun Gao, Wenbo Guo, Erika J Schneider, Zhiyong Lu, Yian Ma, Mark Mueller, Radha Poovendran, Somayeh Sojoudi, Yinglun Zhu, and Dawn Song. Agents' last exam, 2026. URL https://arxiv.org/abs/2606.05405.

Tencent WorkBuddy Bench Team, Siqi Cai, Shaopeng Chen, Xiang Fei, Yong Mao, Zihan Xu, Zhiheng Lyu, Zhijian Shao, Yuchen Shi, Shuwen Zhang, Chaofan Qiu, Linjie Che, Xiaoxi Zhao, Feng Wu, Kai Zhang, Chaofan Zhu, Yubin Qi, Xiaoyun Liang, Peijie Dong, Yunhao Zhang, Yuanjie Zhu, Ling Jiang, Xianjun Zhang, Zhehang Chu, Anyuan Sang, Zhen Feng, Sen Nie, Shi Wu, Yuanzhen Xu, Xin Li, Ning Yang, Zhiqiang Dong, Hande Dong, Qiang Lin, Yi Liu, Yunsheng Wu, Ke Li, and Xing Sun. Tencent workbuddy bench: A multi-domain coding-agent benchmark with contamination-resistant task construction, 2026. URL https://arxiv.org/abs/2607.20911.

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, 2024. URL https://arxiv.org/abs/2407.18901.

Bertie Vidgen, Abby Fennelly, Evan Pinnix, Julien Benchek, Daniyal Khan, Zach Richards, Austin Bridges, Calix Huang, Kanishka Sahu, Abhishek Kottamasu, Bo Ma, Ben Hunsberger, Isaac Robinson, Akul Datta, Chirag Mahapatra, Dominic Barton, Cass R. Sunstein, Eric Topol, Brendan Foody, and Osvald Nitski. The ai productivity index (apex), 2025. URL https://arxiv.org/abs/2509.25721.

Alex Wang, Georg Meinhardt, Jacob Katz, Joseph H. Kim, Pratyush K. Chaudhary, Chase Blagden, and Eric Xu. Bigfinancebench: A workflow-grounded benchmark for financial-research agents, 2026. URL https://arxiv.org/abs/2606.03829.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments, 2024. URL https://arxiv.org/abs/2404.07972.

Frank F. Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z. Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, Mingyang Yang, Hao Yang Lu, Amaad Martin, Zhe Su, Leander Maben, Raj Mehta, Wayne Chi, Lawrence Jang, Yiqing Xie, Shuyan Zhou, and Graham Neubig. Theagentcompany: Benchmarking llm agents on consequential real world tasks, 2025. URL https://arxiv.org/abs/2412.14161.

Lingfeng Zeng, Fangqi Lou, Zixuan Wang, Jiajie Xu, Jinyi Niu, Mengping Li, Yifan Dong, Qi Qi, Wei Zhang, Ziwei Yang, Jun Han, Ruilun Feng, Ruiqi Hu, Lejie Zhang, Zhengbo Feng, Yicheng Ren, Xin Guo, Zhaowei Liu, Dongpo Cheng, Weige Cai, and Liwen Zhang. Fingaia: A chinese benchmark for ai agents in real-world financial domain, 2025. URL https://arxiv.org/abs/2507.17186.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL https://arxiv.org/abs/2306.05685.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents, 2024. URL https://arxiv.org/abs/2307.13854.

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.

A Professional Task Examples

A.1 Finance

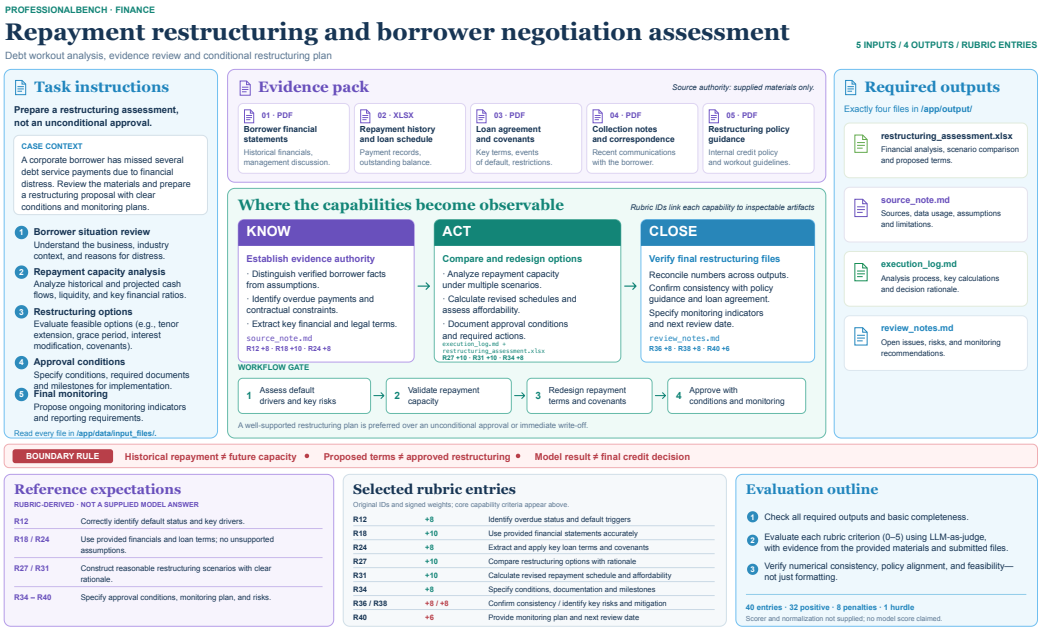


Figure 7: **Finance task example.** A representative finance workflow showing the professional role, task objective, input materials, required deliverables, KAC capability responsibilities, and rubric-based verification criteria.

A.2 Legal



Figure 8: **Legal task example.** A representative legal workflow illustrating how factual materials, applicable rules, professional objectives, deliverables, KAC responsibilities, and evidence-based evaluation criteria are structured within a benchmark task.

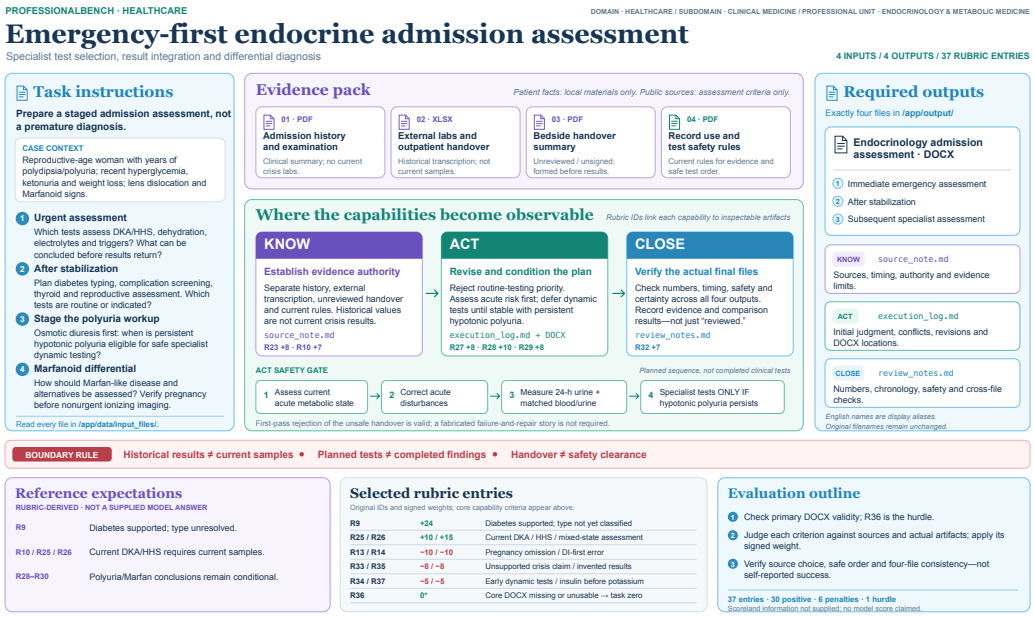


Figure 9: **Healthcare task example.** A representative healthcare workflow demonstrating evidence integration, professional judgment, required outputs, KAC responsibilities, and rubric-based assessment of task completion.

A.3 Healthcare

B Verification Consistency and Reliability

B.1 Expert Review Setup

To validate the reliability of our verification pipeline, we conduct an expert review on a subset of completed task executions. Experts inspect the same artifacts and evidence available to the verifier while remaining blinded to automatic judgments.

The review consists of two complementary evaluations. First, **rubric-level verification** asks experts to independently judge whether each criterion is satisfied using the same binary format as the automatic verifier. This evaluation measures the consistency of fine-grained rubric judgments. Second, **deliverable-level assessment** asks experts whether a completed submission meets the standard of professional delivery, considering correctness, completeness, and usability. This evaluation validates whether the acceptance threshold induced by the verifier corresponds to expert expectations.

B.2 Expert–Verifier Agreement

At the rubric level, automatic verdicts achieve 92.1% raw agreement with expert judgments, indicating high observed consistency in assessing individual criteria. At the deliverable level, experts independently assess submissions that pass automatic verification. Of these submissions, 96.9% are also judged professionally acceptable by experts. This expert-confirmed acceptance rate measures the reliability of automatic acceptance decisions within the reviewed sample. It complements rubric-level agreement by assessing the correctness, completeness, and usability of the deliverable as a whole.

C Implementation Details

C.1 Task Packages and Execution Environments

ProfessionalBench contains 435 executable task packages across three professional domains. Each package is self-contained and includes task instructions, input materials, execution specifications, hidden references, and verification logic. Table 4 summarizes the benchmark scale.

Table 4: **ProfessionalBench task package statistics.**

Statistic	Value
Total executable tasks	435
Finance / Law / Healthcare tasks	141 / 167 / 127
Evaluation subset	90 tasks (30 per domain)
Input files	8,308
Required deliverables	1,880
Rubric items	15,109
Average inputs per task	19.1
Average deliverables per task	4.3
Average rubrics per task	34.7

Each task package contains a solver-visible *instruction.md*, input files under *data/input_files/*, a *task.toml* execution specification, an isolated Docker environment, hidden reference materials, and a *tests/* directory containing verification scripts, *rubric.json*, and optional deterministic checks. Reference solutions are never exposed to the solver and are mounted only during scoring.

We use Harbor 0.22.0 and Docker 29.6.1. Tasks execute in isolated ARM64 Linux containers based on *python:3.12-slim-bookworm*, with task-dependent limits of 1–4 CPUs, 1–8 GiB memory, 2–20 GiB storage, no GPU access, and a 7,200-second agent timeout. Prepared task snapshots, source files, and runtime images are tracked using SHA-256 hashes for reproducibility.

C.2 Models, Harnesses, and Execution Settings

Models are accessed through the Humanlaya gateway using the routes specified in Harbor configurations. Table 5 lists the evaluated models, inference routes, and agent harnesses.

Two route exceptions were retained for exact auditability: healthcare Kimi used *openrouter/moonshotai/kimi-k3*, and law Qwen used *qwen-coder/dashscope/qwen3.8-max-0902*. Codex used high reasoning effort with web search disabled. DeepSeek and GLM used text-only Anthropic-compatible routes. For healthcare and law GLM runs, Claude Code's *Read* tool was disabled, while Bash, Python, PDF extraction, and OCR remained available.

Each task–model pair was executed once, resulting in $90 \times 7 = 630$ primary trials. Runs used two-way concurrency with at most one infrastructure retry. Solver outputs were evaluated immediately by Codex CLI 0.154.0 using *azure/gpt-5.6-sol*. Each rubric item was judged independently with two concurrent judge calls, a 600-second per-rubric timeout, a 7,200-second verifier timeout, and up to three API attempts.

C.3 Reproduction Instructions

Experiments can be reproduced from *ProfessionalBench_Evaluation/official_evaluation*. After installing the validated Harbor version, users should configure the required model providers by specifying API keys, endpoint URLs, model identifiers, and routing names in the provided configuration files. These entries must match the model routes reported in Table 5; otherwise, the resulting runs may not correspond to the reported evaluation setting.

After environment setup, users can audit task packages, execute smoke tests, and launch official runs using the scripts provided in our codebase. The evaluated profiles include *codex-gpt56*, *cc-opus5*,

Table 5: **Model and agent harness configurations.**

Model	Inference Route	Agent Harness
GPT-5.6 Sol	*azure/gpt-5.6-sol*	Codex CLI 0.154.0
Claude Opus 5	*anthropic/claude-opus-5*	Claude Code 2.1.272
Gemini 3.8 Flash	*gemini/gemini-3.8-flash*	Gemini CLI 0.60.0
DeepSeek V4.1 Flash	*volcengine/deepseek-v4-1-flash-260910*	Claude Code 2.1.272
GLM-5.2	*volcengine/glm-5-2-260617*	Claude Code 2.1.272
Kimi K3	*volcengine/kimi-k3-260716*	Kimi Code 2.0.2
Qwen 3.8 Max	*dashscope/qwen3.8-max*	Qwen Code 0.24.1

gemini38-flash, *cc-deepseek*, *cc-glm*, *kimi-k3*, and *qwen38-code*. The archived Harbor configurations, solver trajectories, generated artifacts, verifier logs, per-rubric *verifier/result.json*, and final *result.json* files are the authoritative sources for the reported scores.

D Evaluation Details and Dense Rubrics

This section describes the evaluation environments and dense rubric schema, followed by the scoring rules for signed criteria, dependencies, and hard gates.

D.1 Evaluation Environments Setup

We use two complementary verification methods.

Deterministic programmatic evaluation. Criteria with explicit ground truth are evaluated using executable checks. Examples include comparing spreadsheet values with reference calculations, matching required strings, and checking file existence or validity. Comparisons follow predefined rules, including numerical tolerances where applicable.

Agent-as-a-judge. For criteria requiring semantic or professional judgment, an agent inspects the response and generated deliverables against the rubric. It may consult source materials or execution logs to verify supporting evidence. Each check produces a binary verdict and a brief evidence-grounded justification. Verifier errors are recorded separately rather than treated as failed criteria.

D.2 Rubric Schema

A *criterion* is a self-contained, atomic, binary-verifiable predicate. A *rubric entry* combines the criterion with its annotations and scoring metadata, as summarized in Table 6.

Capability tags. Rubric entries follow the Know–Act–Close framework: **Know** covers professional knowledge, evidence understanding, and groundedness; **Act** covers strategy, reasoning, implementation, and execution boundaries; and **Close** covers completeness, communication, delivery, and actionability. Each entry receives one primary domain and one corresponding subdimension from Table 3. These capability tags are distinct from the workflow *stage*, which records Read, Process, or Output.

Failure attribution. The same framework organizes observed failures, with Infrastructure errors recorded separately. Attribution uses evidence from the execution trajectory and final deliverable rather than automatically converting a failed rubric entry into a causal label. Each observed failure receives at most one evidence-supported primary type; related symptoms are not counted as independent causes.

D.3 Scoring and Dependency Controls

Signed criteria. Let $z_i \in \{0, 1\}$ indicate whether criterion i is true. An ordinary entry contributes

$$\Delta_i = w_i z_i, \tag{2}$$

Table 6: **Dense rubric schema.** Categorical values are shown in English. Enum fields are single-choice; optional dependency and control fields may be omitted.

Field	Type	Definition and values
id	Integer	Stable identifier within the task rubric.
criterion	String	Atomic binary predicate specifying the assessed property and its verdict condition.
stage	Enum	**Read**: interpret instructions and acquire information; **Process**: integrate, calculate, and reason; **Output**: produce the response or deliverables.
target	Enum	**Response**: user-facing reply; **Artifact**: generated files; **Trajectory**: observable operations and execution records.
necessity	Enum	**Explicit**: directly required by the prompt; **Implicit**: necessary for task completion or professional practice; **None**: an auxiliary check not separately assigned to either requirement category.
weight	Signed integer	Importance-weighted credit or penalty: positive values reward desirable properties; negative values penalize failure events.
dimension	Enum	Primary capability domain: Know, Act, or Close.
subdimension	Enum	One capability tag within the selected domain, using the nine subdimensions in Table 3.
info_source	String	Evidence location, such as a prompt clause, document page, spreadsheet range, database table, or execution log.
info_source_detail	String	Specific reference information or justification supporting the check.
depends_on	Integer	Optional predecessor ID; dependency links must be acyclic.
hurdle	Boolean	Hard-gate flag: a true predicate forces the task score to zero. Default: *false*.
bouns	Boolean	Extra-credit flag, tracked separately from the ordinary score. Default: *false*.

where w_i is its signed weight. Thus, a positive-weight entry describes a desirable property, while a negative-weight entry describes a failure event. Entries marked *bouns* receive separate extra credit rather than contributing to the ordinary score.

Dependencies. The *depends_on* links form an acyclic graph connecting related checks. Dependencies do not automatically propagate verdicts: a report and workbook can agree on the same incorrect value, so cross-artifact consistency and numerical correctness remain distinct checks.

Hard gates. An entry marked *hurdle* describes a disqualifying failure. If its predicate is true, the entire task receives zero credit, regardless of other criteria or bonuses. A gate may therefore have weight zero: its effect is a task-level override rather than an additive penalty.

Table 7 illustrates these mechanisms using the attendance-analysis task.

E Failure Diagnosis Protocol

Expert-defined rubrics are inherently goal-oriented: they encode the professional outcomes and capabilities that experts intend to measure. However, rubric-only analysis mainly reveals *which requirements* were not satisfied and may miss latent execution problems or causal mechanisms behind failures. We therefore introduce a complementary failure diagnosis protocol that combines rubric-level requirement analysis with trajectory-based analysis. The former provides structured failure evidence, while the latter reconstructs how failures emerge during execution.

E.1 Rubric-Level Requirement Analysis

Given each task $\times$ run, we first analyze rubric outcomes using simple aggregation statistics. We summarize unmet criteria, failed Professional Gates, and score losses across Requirement Groups, Know–Act–Close domains, and capability tags. Positive-weight criteria are considered unmet when their predicates are not satisfied, while negative-weight criteria are considered violated when their failure conditions are triggered.

Table 7: **Example rubric entries.** Criteria are abbreviated for presentation. Entry 3 records a dependency; entry 5 is a task-level hard gate.

ID	Criterion	Weight	Control
1	Samantha Foster's adjusted unexcused absence count is $10 - 1 = 9$: deduct the January 14 absence covered by approved leave, but not the January 15 partial absence.	$+10$	—
2	The report provides narrative analysis covering all five departments identified in the supplied staff database.	$+6$	—
3	Employee absence counts in the report agree with the corresponding values in the calculation workbook.	$+4$	*depends_on: 1*
4	The response presents fabricated task-specific facts or sources as established evidence.	-8	—
5	Any of the three required deliverables—two PDFs and one XLSX workbook—is missing or cannot be opened.	0	*hurdle: true*
6	The response claims to have checked *greenfield_staff.db*, but a complete execution record contains no supporting access or verification evidence.	-6	—

This step answers *what requirements were not satisfied*, rather than directly identifying the underlying cause. In particular, a failed rubric does not automatically imply a specific failure type: the same unmet criterion may result from a knowledge gap, an execution mistake, or incomplete delivery. Causal interpretation is performed only after trajectory-based analysis.

E.2 Two-Stage Trajectory Analysis

Trajectory analysis separates factual reconstruction from failure classification. For each task $\times$ run, we first generate an **Analysis Card** that compresses the relevant evidence from the task instruction, rubric results, trajectory, grader feedback, final artifacts, and environment logs. The card provides a unified evidence interface for subsequent classification and prevents the classifier from relying on task-specific raw traces.

Each Analysis Card contains:

- task, model, and run identifiers;
- task requirements and key constraints;
- execution summary and correctly completed actions;
- observed errors, affected rubrics, and artifact status;
- evidence pointers to trajectory steps, files, or grader results;
- confidence and unresolved uncertainties.

Stage I does not assign final failure labels. It only reconstructs what happened, where deviations occurred, and how later outcomes were affected.

Stage I Prompt (Analysis Card Generation)
Analyze each task $\times$ run using the available task instructions, rubrics, trajectory logs, grader results, generated artifacts, and execution metadata.
The goal of this stage is evidence-grounded reconstruction, not failure classification. Do not assign labels from the failure taxonomy or infer unobserved internal states. A failed rubric indicates an unmet requirement, not necessarily its underlying cause.
For each run, generate an Analysis Card containing:
(1) **Task Summary**: summarize the objective, key constraints, and required deliverables relevant to the observed outcome;
(2) **Correct Behaviors**: identify important steps, analyses, tool calls, calculations, or artifacts that were completed correctly, with evidence pointers;
(3) **Observed Problems**: describe what went wrong by comparing the expected behavior with the actual execution. For each issue, provide: - expected result or requirement; - observed behavior; - earliest observable deviation; - affected rubric items; - supporting evidence locations;

(4) **Execution and Artifact Status**: summarize final artifact quality, missing outputs, grader results, and relevant trajectory events;
(5) **Uncertainty**: distinguish confirmed errors supported by direct evidence from possible causes that require further validation.
Use concrete evidence references whenever possible, such as trajectory steps, tool calls, artifact locations, rubric IDs, or source sections. When multiple failed rubric items originate from the same upstream mistake, describe the shared cause and downstream effects separately.
Do not classify an error as knowledge, strategy, implementation, or completion failure at this stage. The purpose of the Analysis Card is to provide a standardized evidence interface for a later independent failure classifier.

Stage II performs failure attribution using an independent classifier agent. The classifier receives only the Analysis Card and the predefined failure taxonomy in Table 3; it does not access the original trajectory or artifacts. This design ensures that different tasks and models are compared through a common evidence representation.

The core decision rule is a **Counterfactual Repair Test**:

> *If all other factors remain unchanged, what is the smallest type of repair required to recover the lost performance?*

This criterion distinguishes failures with similar surface symptoms. For example, an incorrect spreadsheet value may correspond to: (i) *Wrong Strategy* when the calculation logic is inappropriate; (ii) *Implementation Bug* when the method is correct but the formula is wrong; or (iii) *Incomplete / Abandoned* when required cells are never completed.

Stage II Prompt (Failure Classification)
You are a failure diagnosis agent. Your task is to classify the primary failure mode of an agent run using only the provided Analysis Card and the predefined failure taxonomy.
Do not access the original trajectory, task files, artifacts, or grader logs. The Analysis Card is the only evidence source.
Use the Counterfactual Repair Test: "If all other factors remain unchanged, what is the minimal type of change required to fix the observed failure?"
When multiple rubric items fail, do not treat every failed rubric as an independent cause. Identify whether several failures are downstream effects of the same upstream problem.
Return a structured JSON output with fields: primary failure, evidence, minimal repair, affected rubrics, secondary failures, uncertainties, and confidence.

E.3 Integrated Failure Localization and Analysis

The final diagnosis combines rubric-level evidence with trajectory-level localization. Each diagnosed failure is linked to: (i) unmet professional requirements; (ii) the earliest supporting trajectory evidence; (iii) the assigned Know–Act–Close failure type; and (iv) affected downstream outcomes.

A single upstream error may explain multiple failed rubric items. We therefore avoid counting correlated symptoms as independent failures. For example, if an incorrect calculation propagates into several reports, the calculation error is treated as the primary failure and the later inconsistencies as consequences.

Human review is applied to ambiguous cases, including conflicting grader feedback, insufficient evidence, and possible infrastructure failures. When evidence supports multiple explanations, the analysis records uncertainty rather than forcing a unique attribution. The resulting diagnosis should be interpreted as evidence-supported post-hoc analysis; without intervention experiments, it does not claim causal validation.

F Detailed Taxonomy and Benchmark Statistics

This section describes the taxonomy and composition of the frozen ProfessionalBench release, including coverage across domains, subdomains, and Professional Units, task and workflow distri-

butions, and statistics on input resources, required deliverables, and rubric items. These statistics characterize the benchmark independently of model performance.

F.1 Full Taxonomy and Instantiated Coverage

Figure 10 presents the professional task hierarchy. Healthcare's broader Professional Unit coverage is largely driven by the detailed Clinical Medicine taxonomy, while finance and law span more subdomains. All categories are counted within domains, with similarly named categories treated as distinct nodes.

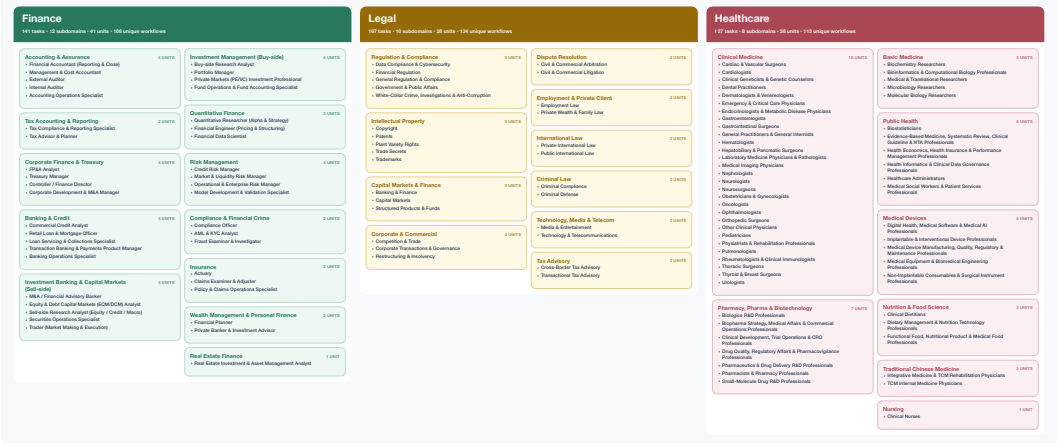


Figure 10: **Full taxonomy and instantiated coverage.** Professional Units organized by domain and subdomain, with domain-level coverage totals.

F.2 Task and Workflow Distributions

Figure 11 distinguishes coverage breadth from workflow density. Law contains more workflows per Professional Unit, whereas healthcare has broader but sparser unit coverage. The subdomains with the most workflows do not always have the highest density: Investment Management (Buy-side) leads in finance and Dispute Resolution in law when normalized by Professional Unit count.

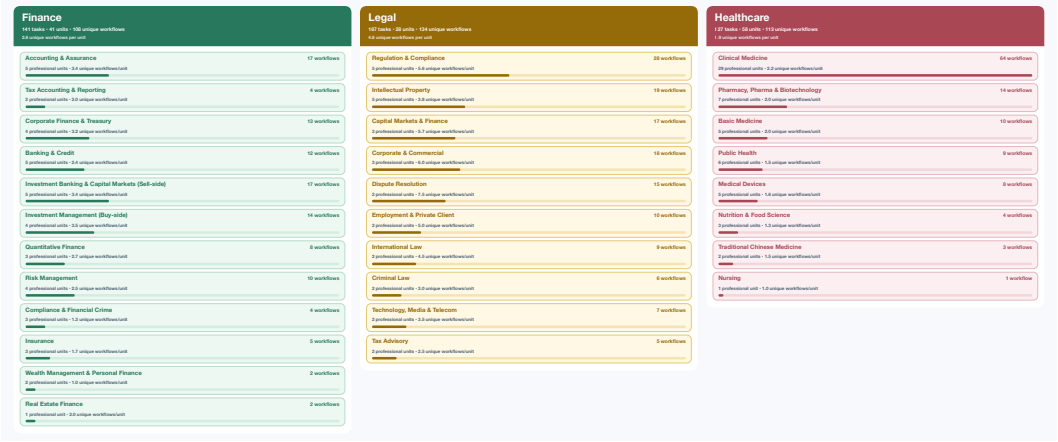


Figure 11: **Task and workflow distributions.** Task totals, distinct Anchor Workflow counts, and workflows per Professional Unit across domains and subdomains.

F.3 Resources, Deliverables, and Rubric Statistics

Figure 12 shows distinct task-composition profiles. Finance has the most input files and required deliverables per task, while healthcare has the most rubric items per task. Law contributes the largest total numbers of deliverables and rubric items, reflecting its larger task pool rather than the highest per-task counts.

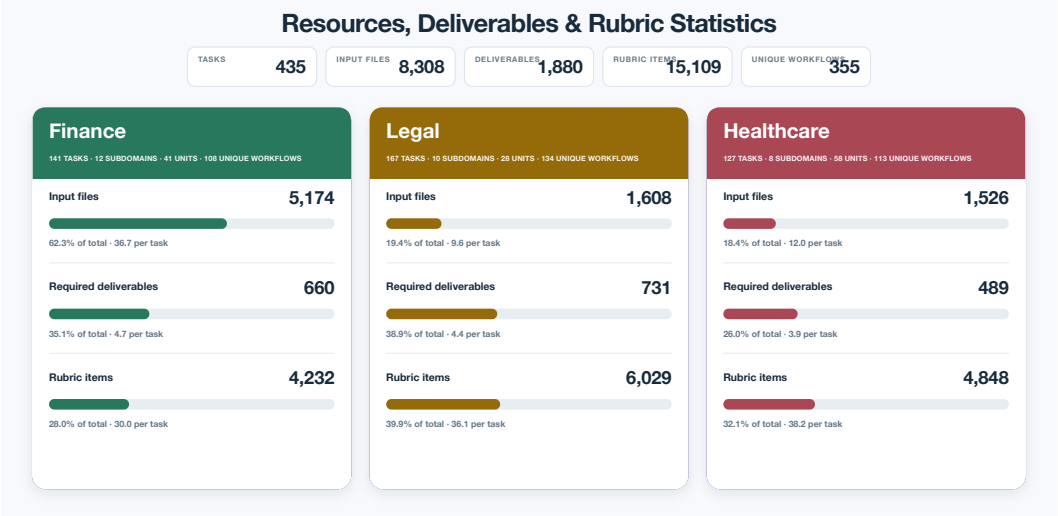


Figure 12: **Resources, deliverables, and rubric statistics.** Input files, required deliverables, and rubric items by domain, with aggregate and per-task statistics.