UpBench: A Dynamically Evolving Real-world Labor-market Agentic Benchmark Framework Built for Human-Centric AI

Darvin Yi   DARVINYI@UPWORK.COM

Teng Liu   TEDLIU@UPWORK.COM

Mattie Terzolo   MATTIETERZOLO@UPWORK.COM

Lance Hasson   LANCEHASSON@UPWORK.COM

Ayan Sinha   AYANSINHA@UPWORK.COM

Pablo Mendes   PABLOMENDES@UPWORK.COM

Andrew Rabinovich   ANDREWRABINOVICH@UPWORK.COM

Abstract

As large language model (LLM) agents increasingly undertake complex, sustained digital work, reliable frameworks are needed to evaluate their real-world competence, adaptability, and capacity for human collaboration. Existing benchmarks remain largely static, synthetic, or domain-limited, providing limited insight into how agents perform in dynamic, economically meaningful environments. We introduce UpBench, a dynamically evolving benchmark grounded in real jobs drawn from the global Upwork labor marketplace. Each task corresponds to a verified client transaction, anchoring evaluation in genuine work activity and financial outcomes. UpBench employs a rubric-based evaluation framework, in which expert freelancers decompose each job into detailed, verifiable acceptance criteria and assess AI submissions with per-criterion feedback. This structure enables fine-grained analysis of model strengths, weaknesses, and instruction-following fidelity beyond binary pass/fail metrics. Human expertise is integrated throughout the data pipeline (from job curation and rubric construction to evaluation) ensuring fidelity to real professional standards and supporting research on human-AI collaboration. By regularly refreshing tasks to reflect the evolving nature of online work, UpBench provides a scalable, human-centered foundation for evaluating agentic systems in authentic labor-market contexts, offering a path toward measuring how AI agents can play a primary role of augmentation rather than automation.

Keywords: Agentic AI · Human-in-the-loop Evaluation · Benchmarking · Large Language Models · Real-world Datasets · Human–AI Collaboration

1 Introduction

Large language model (LLM) agents are increasingly capable of reasoning, planning, and acting in complex environments. As these systems begin to perform sustained digital work, evaluation methods must evolve accordingly. Traditional benchmarks are static, synthetic,

and narrowly scoped, which offer limited insight into how agents perform on tasks that reflect real professional expectations. Progress in agentic systems therefore requires benchmarks that capture the dynamics, variability, and feedback-rich structure of real-world work.

We introduce **UpBench**, a large-scale and continually evolving benchmark designed to evaluate agent performance on economically grounded tasks. Each example in UpBench corresponds to a real job drawn from a global labor marketplace, including historical deliverables and verified client payouts. This foundation anchors evaluation in genuine human demand and practical utility rather than synthetic proxies.

UpBench emphasizes both breadth and depth. The dataset spans diverse professional domains and includes detailed rubrics that decompose tasks into verifiable acceptance criteria. Each AI submission is paired with per-criterion feedback generated by human experts, enabling fine-grained analysis of model behavior beyond binary pass/fail metrics. Because the dataset is refreshed over time, it also reflects the evolving structure of work and skills in the digital economy.

Beyond its use as a benchmark, UpBench provides an environment for studying human–AI collaboration. The data pipeline integrates human expertise at multiple stages (i.e., qualification, rubric creation, and evaluation) allowing systematic examination of how human feedback shapes agent performance and alignment. This human-in-the-loop design supports research on co-evolutionary training, supervision quality, and the role of human judgment in agent development.

Together, these features position UpBench as a scalable foundation for evaluating and improving agentic systems in realistic contexts. By grounding assessment in real economic activity and human feedback, it provides a path toward measuring how AI agents can augment, rather than simply automate, human work.

2 Related Work

2.1 Agentic AI Benchmarks

A growing body of work recognizes that evaluating AI agents requires moving beyond traditional static tasks to agentic settings in which systems plan, act, and deliver outcomes in more realistic work-like environments. For example, the recent benchmark described in "Holistic Agent Leaderboard: The Missing Infrastructure for AI Agent Evaluation" addresses "holistic agent" evaluation across nine models and nine benchmark domains (coding, web navigation, science, customer service), reporting over 21,000 rollouts to analyze interactions among model, scaffold, and benchmark (Kapoor et al. (2025)).

Other general-purpose agent benchmarks such as "AgentBench: Evaluating LLMs as Agents" explore LLM-agent performance in interactive, multi-step decision-making environments (Liu et al. (2023)). More recently, frameworks like "Establishing Best Practices for Building Rigorous Agentic Benchmarks" have identified methodological gaps in benchmark design (hold-out sets, reward design, task specification) and offered a checklist for more rigorous agentic evaluation (Zhu et al. (2025)).

In the sub-domain of economically grounded work evaluation, two notable initiatives are worth highlighting. First, the AI Productivity Index (APEX) from Mercor assesses frontier AI models on tasks drawn from investment banking, consulting, law, and medicine (i.e.,

knowledge-work sectors with direct economic relevance) (Vidgen et al. (2025)). Second, GDPval from OpenAI evaluates model performance on tasks from 44 occupations covering major GDP-contributing sectors, with deliverables mapped to real-world work (Patwardhan et al. (2025)).

These efforts signal a shift toward benchmarking what agents can do in person-work settings, rather than purely synthetic reasoning or code-completion tasks. However, several limitations remain: many benchmarks focus on single-turn deliverables rather than full workflows, may use proxies rather than true client-facing deliverables, and often lack ongoing temporal refresh or rubrics with per-criterion feedback.

2.2 Human-in-the-Loop & Centaur Evaluation Studies

Beyond benchmark design, there is a rich literature on how humans are integrated into evaluation loops—both as collaborators with agents (centaur settings) and as supervisors of agent behaviour. The position paper AI Should Not Be an Imitation Game: Centaur Evaluations argues that evaluation regimes should shift from pure automation (agent alone) to centaur evaluations, in which humans and AI systems jointly solve tasks, enabling measurement of human augmentation, interpretability, and collaboration (Haupt and Brynjolfsson (2025)). Studies of human-in-the-loop systems in machine learning more broadly show, for example, how allowing human feedback may reduce trust in a system even if accuracy improves (Honeycutt et al. (2020)). These works underscore the importance of designing pipelines that respect human-agent interaction, supervision quality, and feedback granularity—not just agent autonomy. For agentic work benchmarks, this means tasks, rubrics, feedback loops, and human–AI orchestration become evaluation design elements.

2.3 Previous Benchmarks Built on Upwork Data

This paper does not represent the first time Upwork's Labormarket data was used in the construction of an agentic benchmark. SWE-Lancer (Miserendino et al. (2025)) evaluates frontier LLMs on real freelance software-engineering projects, measuring their ability to complete tasks that previously resulted in actual client payments. It provides one of the first economically grounded benchmarks for end-to-end coding work. Additionally, the Remote Labor Index (RLI) is a multi-sector benchmark built from real, paid remote-work projects, sourced largely from freelance platforms and organized using the Upwork taxonomy. The RLI aimed to evaluate end-to-end AI agents on economically valuable tasks with full briefs, inputs, and human gold-standard deliverables (Mazeika et al. (2025)).

Additionally, our prior work lays the foundation for the current study. In a prior paper, Towards Real-World Evaluation of Agentic Work in Freelance Marketplaces we introduced a dataset derived from real freelance marketplace jobs, grounding tasks in actual economic transactions. Secondly, in A Scalable Agentic Environment for Real World Work we expanded the pipeline to support multi-agent interaction and a structured data-collection architecture. These contributions established (i) the viability of real job-based benchmarks, (ii) an initial rubric and feedback system, and (iii) a human–AI collaboration pipeline (Terzolo et al. (2025)). The present work builds on those foundations by scaling dataset size, increasing domain diversity, deepening rubric granularity with per-criterion feedback, and introducing a continuously refreshed stream of tasks.

2.4 Our Contributions

Relative to the existing body of work, our paper makes the following major contributions: 1) We use real-world jobs where every task corresponds to a historical job with verified client payout, affording economic grounding and authenticity. 2) We create hand-generated rubrics attached to each job, decomposing requirements into acceptance criteria. 3) We implement human grading of rubrics with feedback, where expert human freelancers grade model submissions along each rubric criterion and provide per-criterion feedback, enabling fine-grained evaluation. 4) We establish a human-centric data-collection and evaluation pipeline that, rather than only evaluating agents, emphasizes human–AI collaboration in data collection, qualification, rubric design, feedback loops and continuous refresh of tasks.

3 Method

Our data pipeline transforms raw marketplace records into a research-grade benchmark suitable for large-scale evaluation of AI agents on real professional work. The pipeline consists of four main components: filtering jobs, additional job categorization, rubric creation, and evaluation of job deliverables.

3.1 Human-Centric Data Collection

A central feature of the UpBench pipeline is its reliance on human expertise for dataset construction, annotation, and evaluation. Rather than relying solely on synthetic supervision or crowd-sourced labeling, we engage verified experts from Upwork's own freelancer network to ensure that each stage of the data-generation process reflects real professional standards.

All contributing experts are active Upwork freelancers who meet strict eligibility criteria. Every recruited expert freelancers have a job satisfaction score of 100% with every freelancer having a top rated or top rated plus badge. Our freelancers have a cumulative earnings of over $1M on the Upwork platform with a combined 96,000 hours on the platform. Experts are contracted directly through Upwork, allowing the data-collection process to leverage the same professional marketplace dynamics that underlie the benchmark itself.

Each expert participates in one or more stages of data collection, including qualification verification, rubric construction, and deliverable evaluation. Jobs are distributed among experts according to their domain specialization, corresponding to eight predefined categories of work:

- Accounting & Consulting
- Admin Support
- Data Science & Analytics
- Engineering & Architecture
- Sales & Marketing
- Translation
- Web, Mobile & Software Development
- Writing

To maintain domain alignment and evaluation consistency, experts are grouped into two broad specialization sets:

Table 1: Expert set assignments by job category. Each set represents a group of Upwork professionals selected for their domain specialization and experience.

Set Name	Associated Job Categories
General	Accounting & Consulting; Admin Support; Sales & Marketing; Translation; Writing
Tech	Web, Mobile & Software Development; Engineering & Architecture; Data Science & Analytics

Within each set, assignments are balanced to ensure both coverage and cross-review. Experts receive domain-matched tasks to construct rubrics, assess deliverables, and provide per-criterion feedback. This organization supports consistent annotation standards while capturing the authentic variation in professional judgment across different fields.

By embedding qualified human professionals directly in the data-collection pipeline, UpBench maintains fidelity to real-world evaluation practices. This design ensures that the benchmark not only reflects genuine job structures and deliverables but also the evaluative perspectives of practitioners who perform such work in production environments.

3.2 Filtering Jobs

We begin from jobs that have been verified to be completed from Upwork. To ensure consistency and replicability, we restrict the dataset to fixed-price, single-milestone jobs with clearly defined requirements and verified client payments. Hourly projects are excluded to avoid ambiguity in scope and outcome quality.

3.3 Additional Job Categorization

After initial filtering, each qualified job was reviewed by domain-matched expert freelancers. They assigned two labels:

- task completeness: whether the job post contained sufficient context to be completed as written
- presence of personally identifiable information (PII): these labels support dataset quality control and analysis of data sensitivity.

For the benchmark project, freelancers did not access historical client–freelancer communications or prior deliverables. All assessments were based solely on job post content and any new materials created for this project, ensuring privacy compliance and a clear separation from prior marketplace activity.

This expert-led labeling adds practical human judgment while maintaining consistency and scalability. The framework also allows new tags or task-specific extensions without changing the overall pipeline. Qualified jobs were then verified by an additional expert, ensuring that each example reflects a real, completed project.

3.4 Rubric Creation

Each job is paired with a rubric—a structured set of acceptance criteria derived from the original job post, attachments, and deliverable context. Rubrics are designed to transform open-ended client instructions into measurable, verifiable requirements that support both automated and human evaluation.

Expert annotators construct between 5 and 20 rubric criteria per job, labeled according to their importance: critical, important, optional, or pitfall (criteria to avoid). This labeling scheme supports graded assessment rather than binary scoring and provides interpretable insights into agent performance. Rubrics are authored manually by domain experts, ensuring precision and domain alignment, and serve as both evaluation scaffolds and interpretive tools for analyzing agent errors.

3.5 Evaluation of Job Deliverables

AI model submissions are evaluated against the rubric criteria. Human evaluators assess the same submissions, providing per-criterion feedback and justifications. The per-criterion feedback, authored by expert freelancers, captures nuanced quality distinctions such as completeness, factual correctness, and alignment with client intent. These structured feedback traces are then stored alongside the job data to enable future research on reinforcement learning from human feedback (RLHF) and dynamic agent retraining.

4 Dataset

4.1 Upwork Labor Marketplace Data

4.1.1 Job Context

The UpBench benchmark comprises 322 real, economically verified jobs from Upwork, the world's largest online labor marketplace where millions of freelancers complete professional contracts across domains such as software development, design, writing, analytics, and consulting. These jobs reflect genuine client demand, measurable financial transactions, and diverse task structures, ranging from short creative briefs to complex technical deliverables. By grounding benchmark tasks in this dynamic labor market, UpBench captures both the breadth of modern digital work and the heterogeneity of skills and evaluation standards required to succeed in it. The benchmark's integration of rubrics, human feedback, and per-criterion grading provides a realistic and scalable framework for assessing how AI agents perform within real-world professional workflows, offering a bridge between experimental model evaluation and the real economics of online work.

Typical Upwork job posts combine a short task summary with explicit constraints that operationalize client intent into concrete work. Posts frequently specify (i) *domain and audience* (e.g., German-language reviews for dating sites; Sinhalese transcription; sustainability-focused eCommerce titles), (ii) *deliverables and format* (e.g., 3D renders, DOCX/PDF reports, spreadsheets, localized text), (iii) *procedural instructions* (bullet lists, step-by-step checks, or acceptance rules such as "6 advantages and 3 disadvantages, ≤ 42 characters each" or "replace invalid emails before payment"), (iv) *tools and platforms* (e.g., Excel, Unity, Mailchimp, Bootstrap, SQL/Python, or data sources like LinkedIn and YouTube),

(v) *scope and timelines* (word counts, quantity targets, or hard deadlines), and (vi) *communication expectations* (attachments, milestone updates, preferred channels). The examples span content creation, translation, transcription, data collection, analytics, design, coding, and expert consulting. In practice, posts embed de facto acceptance criteria such as file types, quantitative thresholds, formatting standards, or quality gates (e.g., email validation), which UpBench extracts and normalizes into structured rubrics with per-criterion grading. This design captures the diversity and realism of digital work while making it systematically evaluable for agent assessment.

4.1.2 JOB METADATA

The benchmark dataset contains contract-level data along with detailed job metadata capturing each project's structure, scope, and categorization. Core fields include identifiers (`contract_id`), temporal markers (`start_ts`, `end_ts`), and job descriptions (`job_title`, `job_description`). Payment information is represented by several fields (`job_amount`, `contract_amount`, `milestone_amount`) depending on the contract type. Additional fields describe the job's complexity and context, such as `expertise_tier`, hierarchical categories (`category`, `subcategory`, `subsubcategory`), and meta indicators like the number of milestones or attachments.

Figure 1 summarizes the distribution of contracts across subcategories within each of the eight job categories represented in UpBench. Contract volumes are highest in Engineering & Architecture, which spans multiple specialized fields—most prominently Civil/Structural Engineering and Electrical/Electronic Engineering—and in Sales & Marketing, where Lead Generation & Telemarketing dominates the category mix. In Writing, most jobs fall under Content Writing and Editing & Proofreading Services, while Admin Support is heavily concentrated in Data Entry & Transcription Services. Within Accounting & Consulting, Financial Planning comprises the majority of contracts, and Translation remains overwhelmingly composed of Translation & Localization Services. Technical domains show broader dispersion: Web, Mobile & Software Development includes substantial shares of Web Development and Scripts & Utilities, with smaller but visible portions of AI Applications and Mobile Development. Similarly, Data Science & Analytics is led by Data Analysis & Testing, supplemented by contributions from AI/ML and ETL subfields.

Overall, the subcategory spread reveals a heterogeneous and economically grounded task portfolio spanning operational, analytical, and engineering work. This diversity underscores UpBench's coverage of real-world digital labor—ranging from language-based deliverables and financial analysis to technical implementation and data-driven problem solving.

(a) Accounting & Consulting

(b) Admin Support

(c) Data Science & Analytics

(d) Engineering & Architecture

(e) Sales & Marketing

(f) Translation

(g) Web, Mobile & Software Dev

(h) Writing

Figure 1: Distribution of subcategories within each of the nine UpBench job categories. Bars show the number of contracts observed per subcategory.
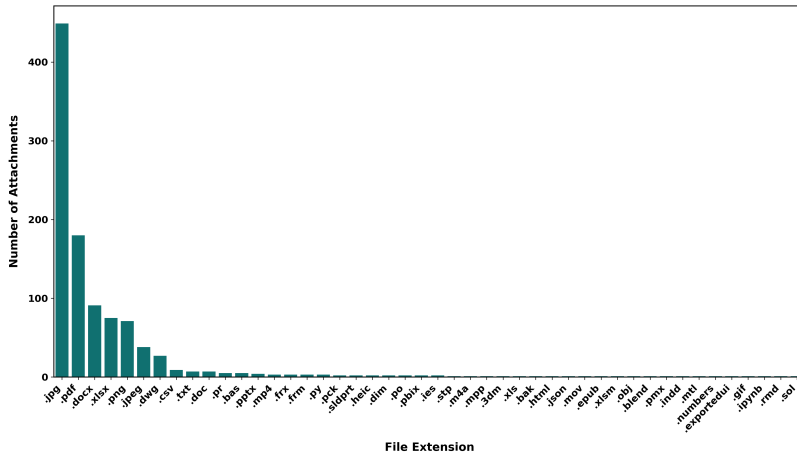
4.1.3 Attachment Distribution

Figure 2: Extensions of Attachment Files

Figure 2 shows the distribution of file types included as job attachments within the filtered UpBench dataset. The most common attachment formats are `JPG`, `PDF`, and `PNG`, followed by document and spreadsheet formats such as `DOCX` and `XLSX`. These patterns suggest that many job posts contain reference materials, design mockups, or textual briefs—consistent with professional service contexts like creative design, analytics, and documentation. The long tail of less frequent extensions (e.g., `.csv`, `.pptx`, `.mp4`) highlights the diversity of file modalities associated with real-world digital work, reflecting heterogeneous job requirements across Upwork categories.

4.2 Rubrics

Each job in the benchmark is accompanied by a rubric—a structured set of acceptance criteria derived directly from the job post, attachments, and deliverable context. These rubrics serve as a bridge between the client's original, often ambiguous instructions and the objective evaluation required for reproducible benchmarking.

Rubric creation is performed by expert freelancers drawn from the relevant job category. Experts decompose the job requirements into 5–20 verifiable criteria, each designed to be specific, measurable, and aligned with the client's stated goals. Criteria are labeled along four importance levels—critical, important, optional, and pitfall—following the labeling convention established in our previous work (Terzolo et al. (2025)).

- *Critical* criteria must be satisfied for a deliverable to be considered acceptable.
- *Important* criteria should be met and reflect strong alignment with the job objectives.
- *Optional* criteria capture desirable but non-essential elements.
- *Pitfall* criteria describe conditions to be explicitly avoided (e.g., factual errors, scope drift, or inappropriate content).

This schema allows fine-grained differentiation of quality across submissions while maintaining consistency across domains. By explicitly mapping the client's goals to verifiable

dimensions of quality, the rubric transforms open-ended professional work into a format suitable for both automated and human evaluation. Full instructions given to freelancers for creating rubrics can be found in Appendix A.

4.3 Submission Feedback

Each submission (whether produced by an AI agent or an AI+human collaboration) is evaluated against its corresponding rubric by a human freelancer. Expert freelancers review the deliverables using the same rubrics. For each criterion, they provide binary scores (pass/fail) along with textual feedback describing observed strengths or deficiencies.

This hybrid setup mirrors the Evaluation Agent paradigm introduced in our previous environment framework (Terzolo et al. (2025)), where evaluators function as externalized reward models. The result is a set of granular, interpretable measurements that allow comparison across models, domains, and supervision modes (AI-only vs. human-in-the-loop).

Rubric-based evaluation ensures consistency and transparency while enabling future extensions, such as per-criterion reward shaping and dynamic feedback loops for iterative agent improvement. Full instructions given to freelancers for evaluating deliverables can be found in Appendix A.

5 Experiments

5.1 Foundational LLM-powered Generalist AI Agents

We evaluate three agent variants that share a common scaffold but differ in the underlying large language model (LLM) used to power them. Each agent operates within the same environment and employs identical prompting, tool use, and feedback protocols. The goal of this setup is to isolate differences in model capability while keeping the surrounding agentic logic and evaluation conditions constant.

All agents follow a simple plan–act–reflect loop inspired by ReAct-style reasoning (Yao et al. (2023)). Given a job description, attachments, and any relevant prior submissions, the agent first parses the task into sub-goals, generates a concise work plan, and then produces a deliverable aligned with the specified rubric. Each agent maintains access to lightweight tools for reading textual attachments and storing generated artifacts. No external retrieval, web access, or fine-tuning is performed during evaluation.

The three model-backed agents used in this study are:

- **Claude Sonnet 4 (Anthropic)**
- **Gemini 2.5 Pro (Google)**
- **GPT-5 (OpenAI)**

Each agent functions analogously to the Worker Agent described in our earlier multi-agent framework (Terzolo et al. (2025)): it receives qualified jobs curated through the UpBench pipeline, produces deliverables, and interacts with the corresponding rubrics and evaluators. The agents are intentionally minimal (no task-specific fine-tuning or external API orchestration), allowing the benchmark to measure intrinsic reasoning, adherence to instructions, and robustness across heterogeneous work domains.

5.2 Key Metrics

5.2.1 COMPLETION RATE

We report two primary quantitative measures of performance: Completion Rate and Rubric Score. We also report several auxiliary metrics that capture the economic and computational implications of agent behavior.

- **Completion Rate** measures the proportion of jobs for which an agent satisfies all required rubric criteria. A job is marked as completed only if every critical and important criterion passes. This strict metric represents full task completion under realistic client standards.
- **Rubric Score** captures partial task completion and provides a graded measure of quality. For each job, the Rubric Score is computed as the fraction of rubric criteria marked "pass": Averaging across all jobs yields a dataset-level measure of how completely agents satisfy task requirements. The Rubric Score complements Completion Rate by quantifying near-misses and incremental improvements that may still represent usable outputs.

To better illustrate the different dimension of successul completion, we define the following system of evaluation and metrics, as seen in Table 2:

Table 2: Definitions of Key Measures

Notations	Definition
N	Total number of jobs **attempted by AI** on the first try ($k = 1$)
A	Number of jobs that **passed** on the first attempt ($k = 1$)
F = N - A	Number of jobs that **failed** on the first attempt
M	Number of **failed jobs** that received a HITL re-attempt ($k = 2$)
B	Number of those M HITL jobs that **passed** after re-attempt
$p_1 = \dfrac{A}{N}$	Pass rate for **AI-only** (first attempt)
$p_{H\|F} = \dfrac{B}{M}$	Conditional pass rate for **HITL among failed jobs**
$p_{\text{overall}} = \dfrac{A + B}{N}$	Observed overall pass rate after adding HITL (lower bound $M < F$)
$R_{j,k} = \dfrac{\#\text{criteria passed}_{j,k}}{\#\text{criteria total}_{j,k}} \in [0, 1]$	Rubric pass score for each job j and attempt k
$\bar{R}_1 = \dfrac{1}{\|N\|} \sum_{j \in N} R_{j,1}$	Average rubric score on first attempt ($k = 1$)
$\bar{R}_{2\|M} = \dfrac{1}{\|M\|} \sum_{j \in M} R_{j,2}$	Conditional rubric pass score for HITL among failed jobs
$\bar{R}_{\text{overall}} = \dfrac{1}{\|N\|} \sum_{j \in N} R_{j,\text{final}}$	where $R_{j,\text{final}} = \begin{cases} R_{j,1}, & \text{if } j \in A \text{ (passed at } k = 1) \\ R_{j,2}, & \text{if } j \in M \text{ (HITL attempt at } k = 2) \\ R_{j,1}, & \text{if } j \in F \setminus M \text{ (failed, no re-attempt)} \end{cases}$

5.2.2 ECONOMIC EFFICIENCY AND VALUE

Beyond these two core indicators, we also report secondary performance metrics that provide additional interpretive context:

- **Economic Value Captured:** By linking each job to its historical payout, we can estimate the real-world monetary value of successfully completed tasks. Summing payout values over successful submissions yields an approximate measure of "earnings potential" for each agent, a proxy for practical economic utility in real labor-market conditions.

- **Runtime Efficiency and Cost:** We compare both median completion time and token-level cost across models to assess the trade-off between performance and computational efficiency. These measures capture the time and resource requirements of each model configuration, supporting a holistic view of performance that integrates quality, speed, and cost.

Together, these metrics provide a comprehensive evaluation of model effectiveness.

5.3 Re-attempts Through Human Feedback

In addition to fully autonomous evaluation, we include a human-in-the-loop (HITL) condition to assess how effectively agents incorporate feedback from expert evaluators. This setting mirrors real-world workflows where humans review, critique, and guide AI-generated work.

The HITL process proceeds in four steps, following the protocol from our prior studies (Terzolo et al. (2025)):

Algorithm 1: Human-in-the-Loop (HITL) Iterative Refinement

Data: job $\mathcal{J}$, rubric $\mathcal{R}$, agent model M, human evaluator H, max iterations K,
cost budget B

Result: Final submission $S^\star$, history $\mathcal{H}$ of attempts and feedback

1   initialize iteration counter $t \leftarrow 0$, cumulative cost $b \leftarrow 0$;
2   initialize submission $S_0 \leftarrow M(\mathcal{J}, \mathcal{R})$;
3   evaluate submission: $(\text{grade}_0, F_0, p_0) \leftarrow H(S_0, \mathcal{R})$;
4   record $(S_0, \text{grade}_0, F_0, p_0)$ in history $\mathcal{H}$;
5   $b \leftarrow b + \text{COST}(\text{grade}_0)$;
6   **while** $p_t = \text{FAIL}$ **and** $t < K$ **and** $b < B$ **do**

     /* Agent revises submission using human feedback   */

7      $t \leftarrow t + 1$;
8      $S_t \leftarrow M(\mathcal{J}, \mathcal{R}, S_{t-1}, F_{t-1})$;
9   evaluate new submission: $(\text{grade}_t, F_t, p_t) \leftarrow H(S_t, \mathcal{R})$;
10   record $(S_t, \text{grade}_t, F_t, p_t)$ in $\mathcal{H}$;
11     $b \leftarrow b + \text{COST}(\text{grade}_t)$;
12   **if** $\text{RUBRICSCORE}(grade_t) \leq \text{RUBRICSCORE}(grade_{t-1})$ **then**
13   **break** ;   /* Optional early stop if no improvement */
14   **end**
15 **end**

We denote each attempt by the AI Agent with the variable k. For example, the first AI-only attempt would be $k = 1$ and the following feedback-fueled re-attempt would be $k = 2$. This is in line with the pass@k formulation from Chen et al. (2021).

This setup allows direct measurement of an agent's responsiveness to human feedback and its capacity for self-correction. Comparing the Completion Rate and Rubric Score between AI-only and HITL conditions quantifies the marginal benefit of human guidance.

Beyond performance improvement, the HITL condition provides structured training data for future agent learning loops—linking per-criterion feedback to behavioral adjustments. It also serves as a diagnostic tool for understanding failure recovery, instruction following, and adaptation in open-ended professional tasks.

5.4 Results

Table 3 and Figure 3 reports performance metrics for three AI agents evaluated under both AI-only (k=1) and Human-in-the-Loop (HITL, k=2) conditions. We additionally show the k=3 HITL loop for our claude model as well as model-based predictions for the k=3 value of the other models (see Appendix D for details). The P_1 column represents the initial success

Table 3: Performance Summary by Agent

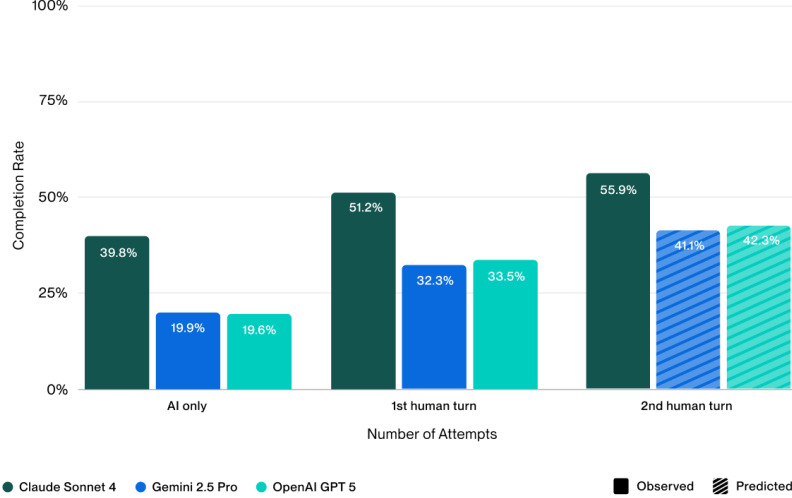| AI Agent	P1	P Overall	Abs Lift	Relative Lift	Rescue Rate
Claude Sonnet 4	0.398	0.512	0.115	0.289	0.233
Gemini 2.5 Pro	0.199	0.323	0.124	0.625	0.180
GPT-5	0.196	0.335	0.140	0.714	0.190

Figure 3: Project Completion Rate: AI Only vs. HITL

rate on first attempt, while P Overall captures the overall success rate after incorporating human feedback and allowing re-attempts. The Absolute Lift (Δp) measures the gain in success rate from adding HITL, whereas the Relative Lift expresses this improvement as a percentage relative to the baseline. The Rescue Rate quantifies the share of initially failed jobs that were successfully "rescued" through human-guided refinement.

Across all three agents, the integration of human feedback leads to substantial performance improvements, exceeding those observed in the initial release. Absolute lifts in success rate now range from 0.115 to 0.140 (11–14 percentage points), corresponding to relative gains between 29% and 71% over the AI-only baseline. The Rescue Rate, which measures the share of initially failed jobs successfully recovered through human-guided refinement, rises to between 18% and 23%—indicating that nearly one in five failed tasks can be salvaged through a single HITL iteration. These stronger gains demonstrate the growing effectiveness and generality of the human-in-the-loop process: even limited expert feedback produces measurable, domain-wide improvements in task success and reliability across heterogeneous model architectures.

Figure 4 compares the average economic values associated with different evaluation settings—AI-only, Human-in-the-Loop (HITL), and aggregate benchmarks. The "Overall Project Value" represents the monetary value of tasks evaluated. The "HITL Cost per Task" bar quantifies the economic price to pay for human-guided refinement. The results
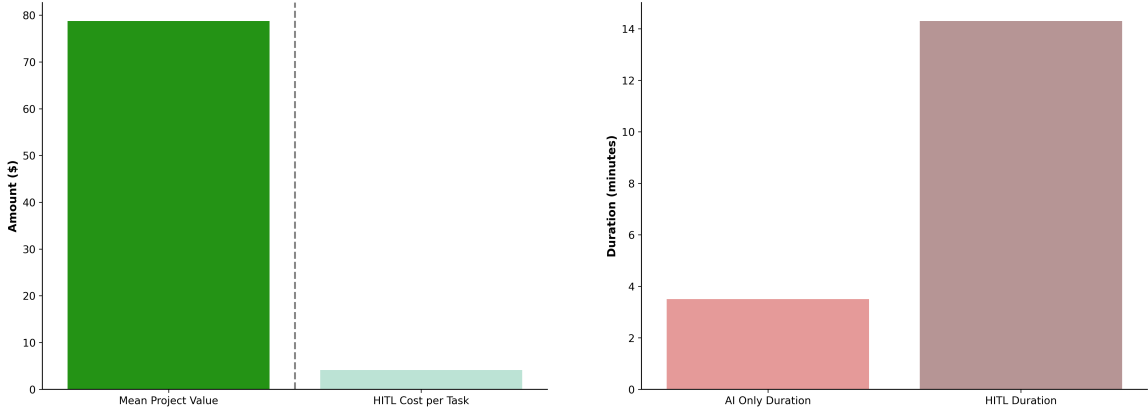
Figure 4: Comparison between project value, HITL cost, and duration.

suggest that while HITL incurs additional cost, it also unlocks measurable economic value, supporting its role in higher-stakes or complex jobs where accuracy justifies the investment.

Figure 5 compares rubric-based performance across the three AI agents, showing average rubric scores before and after the HITL process over the same set of jobs. Across all models, the final rubric scores (purple) consistently exceed baseline AI-only scores (blue), demonstrating that human feedback and re-attempts yield substantial quality improvements. The average absolute rubric lift now ranges from 0.06 to 0.09 on a 0–1 scale—more than double the previous release—representing clear, consistent gains in overall task quality.

The second panel isolates the subset of initially failed jobs that underwent human-in-the-loop re-attempts. Here, each agent exhibits marked improvement after HITL intervention, with relative rubric gains of roughly 40–70%. This conditional view captures the direct effect of human feedback, showing that evaluators help "rescue" a large share of previously unsuccessful attempts. Together, these results reinforce the value of iterative, feedback-driven refinement in the UpBench framework, where even limited human guidance substantially elevates agent performance quality across heterogeneous work domains.

The heatmap Figure 6 summarizes how top-k settings affect agent performance across fine-grained job types. Each row corresponds to a distinct category, while the six data columns show ΔSR for Claude Sonnet 4.0, Gemini 2.5 Pro, and GPT-5 at k=1 and k=2 (left to right within each agent block). Cell color encodes success-rate change. The categories are ordered by average completion rates with higher completion rates at the top.

5.4.1 VALUE COMPARISON

Based on the previous results, we model the expected economic value of completing a task using one of several approaches—AI-only, Human-in-the-loop (HITL), or Human-only—based on their success probabilities, costs, and the value of a successful outcome.

For each approach i, the expected net value is given by:
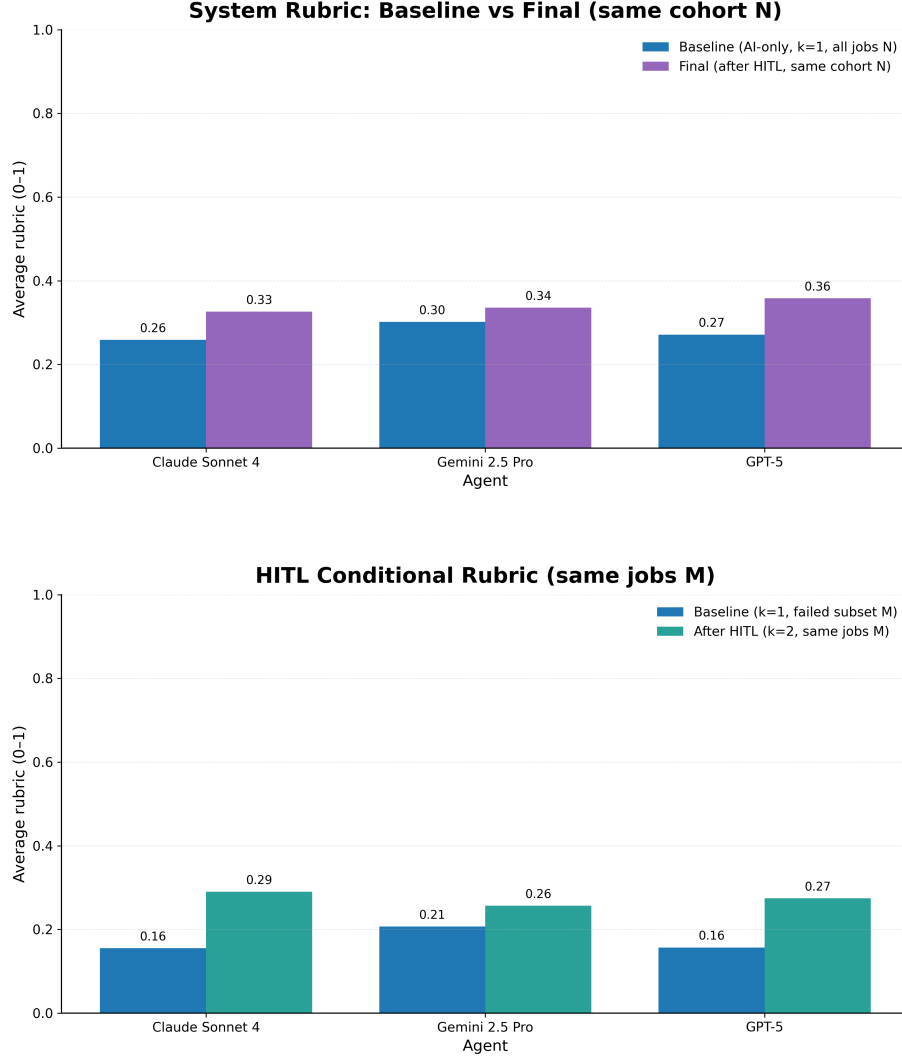
$$E(V_i) = s_i \cdot V - c_i \tag{1}$$

Figure 5: Overall comparison of absolute and relative rubric lift.

where:

- s_i = success rate (probability of success)

- V = economic value of the task if successful

- c_i = cost (time, money, or resources) of that approach

This equation represents the expected net payoff to the buyer or organization for using method i.

The breakeven task value between two approaches, say method 1 and method 2, occurs when their expected net values are equal:
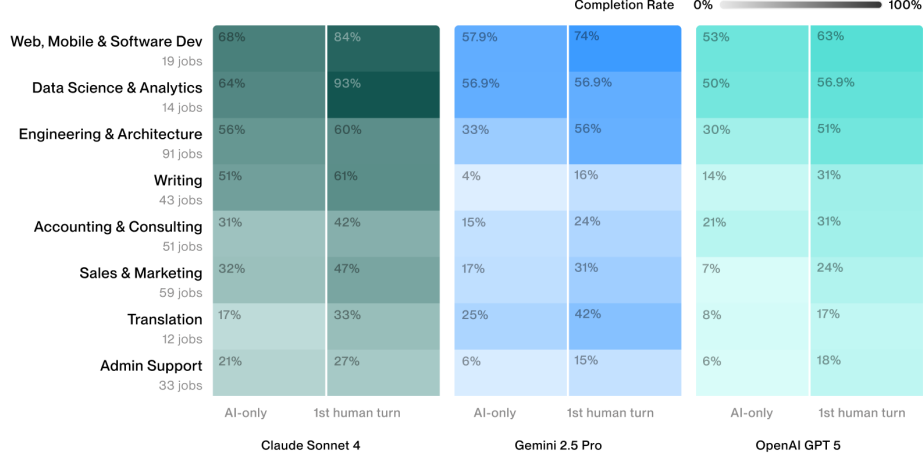
Figure 6: Pixel heatmap of completion rate change (ΔSR) by job categories for three agents.

$$s_H \cdot V^* - c_H = s_L \cdot V^* - c_L$$

Solving for V^* gives:

$$V^* = \frac{c_H - c_L}{s_H - s_L}$$

If $V > V^*$, the higher-success, higher-cost approach (e.g., HITL) yields greater expected net value. If $V < V^*$, the lower-cost, lower-success approach (e.g., AI-only) is more efficient.

Decision Regions Depending on the breakeven thresholds, the optimal approach can be described by:

$$\text{Choose AI-only if} \quad V < V_{\text{AI} \leftrightarrow \text{HITL}}$$

$$\text{Choose HITL if} \quad V_{\text{AI} \leftrightarrow \text{HITL}} \leq V < V_{\text{HITL} \leftrightarrow \text{Human}}$$

$$\text{Choose Human-only if} \quad V \geq V_{\text{HITL} \leftrightarrow \text{Human}}$$

This framework allows direct comparison of cost–performance tradeoffs and helps quantify the economic value of human–AI collaboration.

Figure 7 illustrates the relative economic value of three approaches—AI-only, Human-in-the-Loop (HITL), and Human-only—across tasks of increasing value. At low task values, AI-only solutions yield the highest expected net value because their cost is minimal and the marginal benefit of greater accuracy is small. As task value rises, HITL systems overtake AI by offering higher success rates that justify their added human cost. For very high-value or high-risk tasks, full human execution becomes optimal despite its expense, as the cost of failure outweighs automation gains.

The shaded regions correspond to these regimes: the blue AI-only zone for low-value work, the green HITL zone for mid-value tasks where collaborative feedback adds the most value, and the gold Human-only zone for high-value cases requiring full reliability. The

dashed vertical lines mark the conceptual thresholds where each method's expected net value intersects—AI = HITL on the left and HITL = Human on the right.[1]
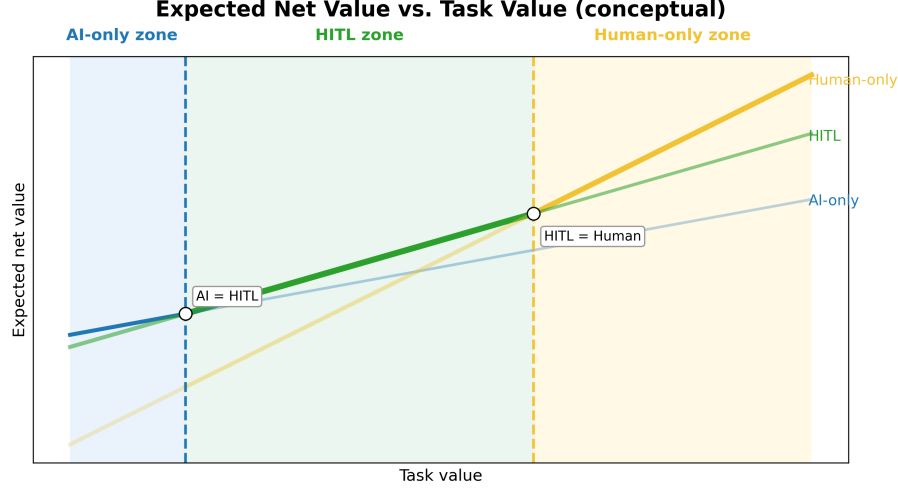

Figure 7: Expected Net Value by Task Value Across AI, HITL, and Human Approaches

5.5 LLM-aided Cluster Analysis

To better understand patterns in rubric design and agent performance, we conduct an LLM-Aided Cluster Analysis over all rubric criteria collected in the UpBench dataset. Each rubric criterion represents a short, natural-language statement describing a specific deliverable requirement (e.g., "the report must include data visualizations" or "the code must run without external dependencies"). Analyzing these at scale provides insight into which classes of requirements are most prevalent and where AI systems tend to underperform.

We begin by embedding all rubric criteria using OpenAI's text-embedding-3-large model, which produces 3,072-dimensional dense representations that capture semantic similarity between textual instructions. These embeddings are normalized and then subjected to hierarchical agglomerative clustering with cosine distance as the similarity metric. The clustering process identifies high-level groupings of criteria that share conceptual similarity (e.g., formatting rules, factual correctness, reasoning requirements). We select the top 20 clusters using distance-based cut thresholds to balance interpretability and coverage across the dataset.

To interpret each discovered cluster, we employ GPT-5 as an analytical assistant. For every cluster, GPT-5 receives the list of rubric criteria and generates a concise textual summary describing the unifying concept and representative examples. This process provides human-readable labels for otherwise latent semantic clusters, producing interpretable themes such as data accuracy, structure and clarity, or adherence to instructions. The

---

1. These boundaries are based on simulation of observed costs and success rates in our sample and should be interpreted as approximate, data-driven thresholds rather than fixed universal values; they will shift with model quality, task type, and human labor cost.

LLM-assisted summaries serve both as descriptive metadata and as anchors for subsequent quantitative analysis.

After clustering, we align each rubric criterion with recorded agent evaluation outcomes. This enables computation of failure frequency per cluster—how often AI agents fail specific categories of criteria relative to others.

(a) Common Criteria Clusters

- Deliverable format and content requirements - 18.45%
- Spreadsheet columns and row requirements - 7.53%
- Language Quality and Formatting Standards - 6.77%
- Automated structured valuation workbook - 5.13%
- Financial report structure and data updates - 3.38%
- Word count & formatting constraints - 2.29%
- Translation accuracy, completeness, and formatting - 1.97%
- Survey response completeness & formatting - 1.97%
- Tailored Resume & Cover Letter Requirements - 1.86%
- Spreadsheet template adherence & completeness - 1.86%

(b) Failed Criteria's Clusters

- Deliverable format and content requirements - 15.49%
- Spreadsheet columns and row requirements - 9.37%
- Financial report structure and data updates - 7.48%
- Automated structured valuation workbook - 6.55%
- Language Quality and Formatting Standards - 5.58%
- UK area K-means maps & outputs - 3.52%
- Task checklist table with six fields - 2.82%
- Translation accuracy, completeness, and formatting - 2.44%
- Word count & formatting constraints - 2.17%
- Event Marketing and Ad Creative Optimization - 2.11%

Figure 8: Criteria Clusters.

6 Limitations

While this study offers a first large-scale evaluation of human-in-the-loop (HITL) performance in agentic AI systems, several limitations should be noted. Below we list some of the strongest limitations for this study.

Rubrics ≠ client acceptance. Our acceptance rubrics emphasize objective, quick-to-verify checks, which do not fully capture qualitative factors (e.g., taste, polish, communication, iteration) that influence real client acceptance.

Minimal agent scaffolding. The evaluated agents were minimal wrappers over foundational LLMs (simple plan–act–reflect; no retrieval, web tools, or fine-tuning), so results are not representative of tool-rich, state-of-the-art agent systems. This study's scope is limited to a few models, domains, and up to two re-attempts.

Restricted contract types. The first release scoped jobs to *fixed-price, single-milestone* contracts, omitting hourly and multi-milestone workflows that add timeline, coordination,

and negotiation complexity. Additionally, our sampling was designed for feasibility rather than to mirror the marketplace-wide job distribution, which can induce category skews and limit generalization. Finally, a corpus of **322** jobs cannot capture the full diversity of Upwork tasks, deliverable modalities, and difficulty levels. We hope to expand the benchmark to capture more complex jobs that better represent all of the jobs on the Upwork Marketplace.

Inter-rater variability unmeasured. While emphasizing consistency and replicability, the study does not yet quantify inter-rater agreement or report error margins. Future work should include multiple independent graders, estimate inter-rater reliability, and present confidence intervals to capture evaluation uncertainty.

HITL lift lacks AI-only ablation. We did not run matched *AI-only* second attempts on the same failed jobs, so the benefit of human feedback cannot be isolated from a "second-chance" effect.

Evaluator expertise profiling. While evaluators are high-performing freelancers, we did not collect fine-grained skill matrices or study how evaluator background moderates judgments or feedback utility.

Strict completion definition. A job is "completed" only if *all critical and important* criteria pass—a deliberately strict standard that may under-recognize partially usable outputs and process quality (e.g., clarifying questions).

Limited exploration of retry depth. Most agents were evaluated only up to $k=2$ (with $k=3$ only sparsely), limiting learning-curve analyses.

Economic modeling is illustrative. The expected-value analysis uses observed costs and simplifying assumptions; it omits several operational frictions and risk adjustments.

Clustering methodology. The LLM-aided clustering pipeline (single embedding model, HAC, LLM-generated labels) may be sensitive to embedding choice, thresholds, and labeler bias.

Together, these limitations suggest the reported results are conservative estimates of achievable performance. Future work will address these gaps by adding AI-only controls, measuring evaluator reliability, and expanding the dataset for greater robustness and replicability.

7 Conclusion

This work introduces UpBench, a novel benchmark that grounds agentic AI evaluation in real-world, economically verified tasks. By leveraging authentic labor-market data from the Upwork platform, UpBench captures the complexity, variability, and professional standards inherent in digital work. Unlike static or synthetic datasets, it reflects the true structure of human demand and provides a rigorous, economically meaningful foundation for measuring AI performance across diverse domains.

Beyond serving as a benchmark, UpBench establishes a continually evolving data-collection framework that integrates expert human oversight throughout every stage—from job selection and rubric construction to feedback generation and evaluation. This dynamic process ensures that the benchmark remains current with the changing nature of work and supports the study of adaptive, feedback-driven AI systems. Ultimately, UpBench represents a commitment to human–AI collaboration, offering not only a tool for measur-

ing performance but also a testbed for understanding how human expertise and agentic intelligence can co-evolve. By embedding human judgment, feedback, and context into the benchmarking process itself, UpBench advances the goal of building AI systems that augment rather than replace human capability in the digital economy.

8 Acknowledgements

References

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, and et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL `https://arxiv.org/abs/2107.03374`.

Andreas Haupt and Erik Brynjolfsson. Position: Ai should not be an imitation game: Centaur evaluations. In *Forty-second International Conference on Machine Learning Position Paper Track*, 2025.

Donald Honeycutt, Mahsan Nourani, and Eric Ragan. Soliciting human-in-the-loop user feedback for interactive machine learning reduces user trust and impressions of model accuracy. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 8, pages 63–72, 2020.

Sayash Kapoor, Benedikt Stroebl, Peter Kirgis, Nitya Nadgir, Zachary S Siegel, Boyi Wei, Tianci Xue, Ziru Chen, Felix Chen, Saiteja Utpala, et al. Holistic agent leaderboard: The missing infrastructure for ai agent evaluation. *arXiv preprint arXiv:2510.11977*, 2025.

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. *arXiv preprint arXiv:2308.03688*, 2023.

Mantas Mazeika, Alice Gatti, Cristina Menghini, Udari Madhushani Sehwag, Shivam Singhal, Yury Orlovskiy, Steven Basart, Manasi Sharma, Denis Peskoff, Elaine Lau, et al. Remote labor index: Measuring ai automation of remote work. *arXiv preprint arXiv:2510.26787*, 2025.

Samuel Miserendino, Michele Wang, Tejal Patwardhan, and Johannes Heidecke. Swe-lancer: Can frontier llms earn \$1 million from real-world freelance software engineering?, 2025. URL `https://arxiv.org/abs/2502.12115`.

Tejal Patwardhan, Rachel Dias, Elizabeth Proehl, Grace Kim, Michele Wang, Olivia Watkins, Simón Posada Fishman, Marwan Aljubeh, Phoebe Thacker, Laurance Fauconnet, et al. Gdpval: Evaluating ai model performance on real-world economically valuable tasks. *arXiv preprint arXiv:2510.04374*, 2025.

Mattie Terzolo, Teng Liu, Darvin Yi, Karthik Gomadam, Lance Hasson, Ayan Sinha, Pablo Mendes, and Andrew Rabinovich. Towards real-world evaluation of agentic work in freelance marketplaces. In *Proceedings of the 39th Conference on Neural Information Processing Systems (NeurIPS 2025)*. Upwork Inc., 2025.

Bertie Vidgen, Abby Fennelly, Evan Pinnix, Chirag Mahapatra, Zach Richards, Austin Bridges, Calix Huang, Ben Hunsberger, Fez Zafar, Brendan Foody, et al. The ai productivity index (apex). *arXiv preprint arXiv:2509.25721*, 2025.

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`.

Yuxuan Zhu, Tengjun Jin, Yada Pruksachatkun, Andy Zhang, Shu Liu, Sasha Cui, Sayash Kapoor, Shayne Longpre, Kevin Meng, Rebecca Weiss, et al. Establishing best practices for building rigorous agentic benchmarks. *arXiv preprint arXiv:2507.02825*, 2025.

Appendix A. Instructions Given to Freelancers

A.1 Instruction for Context Complete Labeling

Definition: A project meets this criterion if all necessary information to complete the work is clearly provided and accessible within the job description and attachments, with no missing details or inaccessible resources.

Requirements: - All necessary information is contained within attachments - Task does not require physical presence or location-based activities - NO substantial follow-up questions needed to complete the work - **Important:** Only consider job description and attachments - DO NOT evaluate deliverables for this criterion

Common Problems to Check: - **Document Count:** If specific numbers mentioned, verify exact count provided (not just examples) - **External Access:** Work requires unavailable external accounts/systems (Shopify admin, proprietary databases) - **Missing Details:** Generic tasks lack specific company/industry/scenario details - **Broken Links:** Referenced links or resources are inaccessible - **Templates vs. Materials:** Attachments are examples to create, not materials to work with - **Essential Information:** Missing product names, titles, subjects, or other completion details

A.2 Instruction for PII Labeling

Definition: A project meets this criterion if all personal, company, and client identifiers are not present in the attachments or job title/description

Requirements: - No personal identifying information (PII) present - Company/organization identity is not discernible - Job poster identity is protected - Any remaining identifiers are appropriately anonymized - **Important:** Only consider job description and attachments - DO NOT evaluate deliverables for this criterion

A.3 Instruction for Rubric Creation

Purpose: After qualifying a project, develop 5-20 objective acceptance criteria that a client could use to evaluate whether a freelancer's work submission meets the contract requirements.

** Requirements** Your criteria must be: - **Objective & Measurable:** Based on verifiable facts, not subjective opinions - **Specific & Concrete:** References exact deliverable requirements from project description - **Clear & Unambiguous:** Anyone should be able to check compliance easily - **Quick to Verify:** Each criterion should be able to be verified in less than 1-2 minutes - **Essential Elements:** Covers key requirements that define successful completion - **Concise:** Should read like bullet points and be brief **Important:** Do not include anything about requested timelines/dates or any attributes about the worker being requested (Ie ignore things about expertise in certain domain or geographic location)

*** Format Guidelines*** The acceptance criteria cell in the worksheet should contain each acceptance criteria as a line with the following structure: "' label - criteria text label - criteria text ... "'

** Possible values for 'label'** - 'critical' - The criterion is critical to the project and must be met for the project to be accepted - 'important' - The criterion is important to the project and should be met for the project to be accepted - 'optional' - The criterion is optional and can be met for the project to be accepted - 'pitfall' - The criterion is a pitfall that must be avoided for the project to be accepted. This is different from the other types in that it is a negative attribute we want to ensure projects do not have.

** Guidelines for 'criteria text'** - Use active, definitive language ("Contains...", "Includes...", "Demonstrates...") - Avoid subjective terms ("high-quality", "professional", "adequate") - Leave empty if no discernible acceptance criteria exist

Examples of Good Acceptance Criteria - "Deliverable contains analysis of all 12 months of data as specified in the dataset" - "Final report includes 5 strategic recommendations with supporting rationale" - "All visualizations are properly labeled with titles, axis labels, and data source citations" - "Must contain between 90-120 words"

** Examples of Poor Acceptance Criteria** - "Work is of professional quality" *(subjective)* - "Analysis is thorough and insightful" *(unmeasurable)* - "Analysis must be completed by end of the week" *(timeline-related)* - "Must be completed by U.S based worker" *(worker-related)*

A.4 Instruction for Deliverable Evaluation

** Instructions for the Evaluation Task**

You are evaluating submitted work deliverables against objective acceptance criteria to determine whether they meet the success criteria established for each project.

** Main Task: Submission Evaluation**

Evaluate submitted work deliverables against the acceptance criteria established during project qualification to determine if submissions meet contractual requirements.

** Evaluation Criteria Application**

For each acceptance criterion provided, you must evaluate whether the submission meets that specific requirement:

pass: The submission clearly and demonstrably meets the criterion

skip: The criterion cannot be properly evaluated or is not applicable Common reasons to skip: - Criterion not relevant to what was requested in the job post - Do not have the tools or skills to evaluate the criterion

fail: The submission does not meet the criterion

Important Guidelines: - Base judgments solely on objective, verifiable facts - Reference specific deliverable content when making assessments - Use the exact acceptance criteria provided - do not modify or interpret them - Each criterion should be evaluated independently - Focus on contract compliance, not work quality beyond the stated criteria

Appendix B. Data Variable Details

B.1 Database Schema Overview

The database consists of six main tables: `projects`, `qualifications`, `submissions`, `evaluations`, `evaluators`, and `agents`. Each table records a different part of the benchmarking workflow.

Table 4: Project Metadata (`projects` table)

Variable	Description
`contract_id`	Unique identifier for each job or contract.
`start_ts`, `end_ts`	Start and end timestamps for the contract.
`job_title`, `job_description`	Title and textual description of the job.
`job_amount`, `contract_amount`, `milestone_amount`	Alternative payment amount fields.
`expertise_tier`	Job difficulty or expertise level.
`category`, `subcategory`, `subsubcategory`	Hierarchical classification of job sectors.
`num_openings`, `num_milestones`, `automation_desire_rating`	Numeric meta fields describing milestones and automation potential.
`onet_tasks`	O*Net task references for the job.
`attachments`	JSON list of attachments linked to the project.

Table 5: Qualification Data (`qualifications` table)

Variable	Description
`qualification_id`	Unique identifier for each qualification event.
`contract_id`, `evaluator_id`	Job and evaluator references.
`qualification_ts`	Timestamp of the qualification event.
`criterion_1--3_judgment/reasoning`	Ratings for context completeness, deliverable quality, and PII absence.
`criterion_4`, `criterion_5`	Additional or experimental criteria.
`acceptance_criteria`	JSON rubric defining evaluation standards.
`is_successful_run`	Indicates whether the qualification process completed successfully.

Table 6: Submission Data (`submissions` table)

Variable	Description
submission_id	Unique identifier for each submission.
contract_id, agent_id	References to the job and submitting agent.
submission_ts	Timestamp of the submission.
deliverables, notes	Submitted content and any accompanying notes.
k	Attempt number (e.g., first, second, etc.).
duration	Duration of the attempt in seconds.
is_successful_run	Indicates if the submission was successfully processed.

Table 7: Evaluation Data (`evaluations` table)

Variable	Description
evaluation_id	Unique identifier for each evaluation event.
contract_id, submission_id, evaluator_id	Links to job, submission, and evaluator records.
submission_feedback	JSON rubric-based feedback for the submission.
judgement	Overall evaluation outcome (pass/fail).
is_successful_run	Indicates if the evaluation process completed successfully.

Table 8: Evaluator and Agent Metadata (`evaluators` and `agents` tables)

Variable	Description
evaluator_id, agent_id	Unique identifiers for evaluator and worker agents.
registration_ts	Timestamp of registration.
is_human	Boolean flag for human or AI status.
name	Name of the evaluator or agent (e.g., Rita, darvin-ms-eval-gpt-5).

Appendix C. Metrics

Lift Calculations for Success Score

1. Absolute Lift (percentage points)

$$\Delta p = p_{\text{overall}} - p_1 = \frac{A + B}{N} - \frac{A}{N} = \frac{B}{N}$$

Interpretation: Increase in overall pass rate (in percentage points) due to adding HITL.

2. Relative Lift (percent improvement)

$$\text{Relative Lift} = \frac{p_{\text{overall}}}{p_1} - 1 = \frac{B}{A}$$

Interpretation: Percent improvement in success rate compared to AI-only baseline.

3. Rescue Rate (conditional on failures)

$$\text{Rescue Rate} = p_{\text{H—F}} = \frac{B}{M}$$

Interpretation: Among jobs that failed initially **and** received a HITL re-attempt, the fraction successfully "rescued" by HITL.

Lift Calculations for Rubric Score

1. Absolute overall lift (rubric points)

$$\Delta \bar{R} = \bar{R}_{\text{overall}} - \bar{R}_1,$$

where $\bar{R}_1 = \frac{1}{|N|} \sum_{j \in N} R_{j,1}$ and $\bar{R}_{\text{overall}} = \frac{1}{|N|} \sum_{j \in N} R_{j,\text{final}}$ with $R_{j,\text{final}} = R_{j,1}$ if $j \in A$, $R_{j,2}$ if $j \in M$, $R_{j,1}$ if $j \in F \setminus M$. (Units: rubric points on $[0,1]$. If reported in percentage points, multiply by 100.)

2. Relative lift among HITL jobs (percent improvement) Define the baseline and post-HITL means on the same failed-and-retried set M:

$$\bar{R}_{1|M} = \frac{1}{|M|} \sum_{j \in M} R_{j,1}, \qquad \bar{R}_{2|M} = \frac{1}{|M|} \sum_{j \in M} R_{j,2}.$$

Then the HITL-conditional relative lift is

$$\text{RelLift}_{H|F} = \frac{\bar{R}_{2|M}}{\bar{R}_{1|M}} - 1.$$

This is the graded analogue of the rescue-rate view: it measures the percent improvement in rubric quality among the jobs that actually received HITL. (If $|M| = 0$ or $\bar{R}_{1|M} = 0$, this quantity is undefined.)

Appendix D. Modeling

D.1 Modified `pass@k` for Real-World Agent Jobs

We evaluate an AI Agent's ability to complete real jobs using a modification of the standard `pass@k` metric for functional correctness of code generation (Chen et al., 2021). For each task i in an evaluation suite $\mathcal{P}$, we generate $n_i \geq k$ independent attempts and observe $c_i \leq n_i$ successful attempts (i.e., attempts that pass validation/tests). The unbiased estimator of `pass@k` for task i is

$$\widehat{\text{pass}}_i(k) \; = \; 1 \; - \; \frac{\binom{n_i-c_i}{k}}{\binom{n_i}{k}}, \qquad \text{with the convention that } \widehat{\text{pass}}_i(k) = 1 \text{ if } n_i - c_i < k, \quad (2)$$

and the aggregate score averages this quantity over tasks:

$$\widehat{\text{pass}}(k) \; = \; \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \widehat{\text{pass}}_i(k). \tag{3}$$

Equation (2) is preferred over the naive estimator $1 - (1 - \hat{p})^k$ (with $\hat{p} = c_i/n_i$), which is biased (Chen et al., 2021, App. A). In practice, to avoid numerical instability when evaluating ratios of binomial coefficients, we compute (2) using the product identity

$$\widehat{\text{pass}}_i(k) \; = \; 1 \; - \; \prod_{j=n_i-c_i+1}^{n_i} \left(1 - \frac{k}{j} \right), \quad \text{and if } n_i - c_i < k \text{ then } \widehat{\text{pass}}_i(k) = 1, \tag{4}$$

which matches the stable implementation recommended in Chen et al. (2021).

Motivation for a ceiling parameter.   In the original formulation, $\widehat{\text{pass}}(k)$ can approach 1 as the number of allowed attempts k grows, provided at least one correct attempt exists among the samples for each task. In real job settings, however, exogenous constraints can preclude perfect success even with many retries. Examples include:

- limited or missing access to required external tools or data sources;

- runtime, budget, or rate-limit constraints on the agent;

- safety, policy, or environment restrictions that prevent specific actions;

- agent-specific limitations (e.g., weak tool-use skills) that systematically reduce feasibility.

Our modification.   To reflect these constraints, we introduce an agent- and suite-specific ceiling parameter $p_{\max} \in [0, 1]$ that caps the attainable pass rate. Our modified metric is

$$\widehat{\text{pass}}^{(\text{agent})}(k) \; = \; p_{\max} \, \widehat{\text{pass}}(k) \; = \; \frac{p_{\max}}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \left(1 - \frac{\binom{n_i-c_i}{k}}{\binom{n_i}{k}} \right). \tag{5}$$

This multiplicative ceiling makes the asymptotic behavior explicit: $\lim_{k \to \infty} \widehat{\text{pass}}^{(\text{agent})}(k) = p_{\max}$, rather than 1. Intuitively, $p_{\max}$ summarizes the fraction of tasks in the suite that are *operationally feasible* for the agent under its tool set, policies, and resource limits. We treat $p_{\max}$ as agent-specific, with the expectation that stronger agents (e.g., better models and/or more capable tool stacks) achieve larger $p_{\max}$.

Modeling choices. In this work we adopt the following choices:

1. Use the unbiased estimator in (2) with the stable computation (4) (Chen et al., 2021).

2. Interpret k as the per-task retry budget consistent with real deployment constraints.

3. Average over tasks as in (3).

4. Apply a single $p_{\max}$ per agent and evaluation suite as in (5) to capture suite-level feasibility constraints; we expect more capable agents to have higher $p_{\max}$.

An alternative (not used here) would be to allow task-dependent ceilings $p_{\max,i}$ to encode known feasibility attributes for specific tasks; in that case $p_{\max}$ in (5) is replaced by $p_{\max,i}$ inside the sum.

D.2 Model Fitting

With n, c, and $p_{\max}$, equation 5 has 3 degrees of freedom. In our collected data, we have calculated pass rates at k=1 and k=2 for 3 of our agents and a k=3 (2nd HITL re-attempt) for one of our agents. Just to get rough modeling numbers, we fit equation 5 on our 3 data point agent to find an estimate for $p_{\max}$. We share this value for the other two agents' model but fit n and c separately. We use this to generate a rough prediction of the other two agents' performance at k=3.

Through this process, we find a fit value of $p_{\max} = 0.57$.

D.3 Limitations

While the capped `pass@k` in Eq. (5) is a useful abstraction for real-world agent performance, our current fit has several important limitations.

(1) Data scarcity and identifiability. We fit three free parameters $(n, c, p_{\max})$ with effectively three observed points (aggregate pass@k at $k \in \{1, 2\}$ for all agents and $k=3$ for one agent). This saturated setting is highly ill-conditioned: small measurement noise can lead to large shifts in the fitted parameters, and multiple $(n, c, p_{\max})$ triplets can produce near-indistinguishable curves over $k \in \{1, 2, 3\}$. As a result, the $k = 3$ predictions for the two agents without $k = 3$ observations should be treated as high-variance extrapolations rather than precise forecasts. Moreover, because we fit on aggregated pass rates, n and c should be interpreted as *effective* parameters rather than literal per-task sample counts, and heterogeneity across tasks can further blur their meaning. A larger dataset—with more tasks, more k levels (e.g., $k \geq 3$ for all agents), and repeated measurements—would materially reduce variance and improve parameter identifiability.

(2) Shared ceiling parameter $p_{\max}$. We transfer a single fitted ceiling ($p_{\max} = 0.57$) from the agent with $k = 3$ data to the other agents. In practice, $p_{\max}$ is induced by constraints that are *agent–suite–environment* specific (tool availability, rate limits, budgets, safety policies, orchestration quality). Assuming a common ceiling across agents is unlikely to hold and can bias relative comparisons: stronger agents (or richer tool stacks) may legitimately attain higher ceilings, while agents with tighter constraints may have lower ceilings. Future analyses should allow $p_{\max}$ to vary at least by agent (and potentially by

evaluation suite), with partial pooling or hierarchical priors to share signal without forcing equality.

(3) Non-IID re-attempts and feedback dependence. The standard `pass@k` estimator presumes attempts are i.i.d. samples from the same proposal distribution. Our setting explicitly violates this assumption: later attempts are constructed using submission feedback and (for $k = 3$) a human-in-the-loop re-attempt. These dependencies typically *increase* success probabilities relative to i.i.d. sampling and break the unbiasedness guarantees of the classical estimator. Consequently, our reported curve is best interpreted as a *policy-dependent* "pass-within-k" under our specific feedback protocol, not as the canonical i.i.d. `pass@k`. Direct comparison to literature values computed under i.i.d. sampling may therefore be misleading.

Implications and recommended mitigations. To address the above, we plan to (i) expand data collection to include more k levels per agent and more tasks, (ii) report uncertainty (e.g., bootstrap across tasks or Bayesian posteriors) for $p_{\max}$ and the predicted pass@k, (iii) estimate $p_{\max}$ per agent (and/or per suite) with partial pooling rather than sharing a single value, and (iv) complement `pass@k` with a policy-aware sequential metric (e.g., empirical "pass within k" under our workflow, or a simple hazard/learning-curve model) to faithfully capture feedback-driven retries.