UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

DÉPARTEMENT DES SCIENCES ADMINISTRATIVES

WORKING PAPER NO. 42

The Autonomous White-Collar Worker

Measuring the Economic Frontier of Claude Code, Computer-Using Agents, and Autonomous Knowledge Work

Simon-Pierre Boucher

Département des sciences administratives, Université du Québec en Outaouais

simon-pierre.boucher@uqo.ca

283, boulevard Alexandre-Taché, Gatineau (Québec) J8X 3X7, Canada

November 2026

Version 1.0

Abstract

Keywords: autonomous agents; computer-use agents; Claude Code; task-based automation; labor economics; experimental economics; benchmarks; reliability; cost-effectiveness

JEL Classification: C93; J24; O33; L86; M15; C88

Contents

1 Related literature **5**

1.1 Task-based models of automation and occupational exposure 5
1.2 Field and laboratory experiments on generative-AI productivity 7
1.3 Agent benchmarks and computer-use evaluations 8
1.4 The economics of AI agents, delegation and supervision 10
1.5 Reliability, long-horizon failure, self-repair and judge validity 13
1.6 Experimental methodology 14
1.7 Positioning: precedents and incremental contribution 15

2 A task-based framework for autonomous white-collar work **17**

2.1 Occupations, tasks and runs 17
2.2 The Autonomous Labor Frontier 19
2.3 The Computer-Use Tool Premium 20
2.4 The Workflow Dependency Frontier and the Autonomous Recovery Index 22
2.5 Autonomy-Adjusted Productivity 24
2.6 The Professional Reliability Frontier 25
2.7 Delegation with verification: a principal–agent reading 25
2.8 What the frontier is not 27

3 The occupational benchmark and the Virtual White-Collar Economy **28**

3.1 The occupational taxonomy 28
3.2 The Virtual White-Collar Economy 28
3.3 Anatomy of a scenario 30
3.4 Generation and independent validation 32
3.5 The pilot: ten scenarios in ten occupations 33
3.6 Scaling the benchmark 37

4 Experimental design **37**

4.1 Treatment arms 37
4.2 Information constancy, randomisation, interleaving and repeats 38
4.3 Caps 40
4.4 Measurement framework 40
4.5 Outcome definitions 41
4.6 The evaluation pipeline and its blinding 42

4.7 Preregistered hypotheses . . . 43
4.8 Estimands . . . 43
4.9 Inference plan . . . 44
4.10 Scale decision rule . . . 44
4.11 Exclusions and missingness . . . 45
4.12 Power simulation . . . 45
4.13 The human arms: designed, not executed . . . 46

5 Laboratory infrastructure: Claude Code calling Claude Code 47
5.1 Controller and executors . . . 47
5.2 The measured environment . . . 48
5.3 The launcher . . . 49
5.4 Workspace isolation, portal, office and browser adapters . . . 49
5.5 The macOS layer: what works headlessly and what does not . . . 50
5.6 The A4 self-repair protocol and bounded recovery . . . 51
5.7 Logging, hashing, ledger, dashboard and emergency shutdown . . . 51
5.8 Scheduling and concurrency . . . 52

A Mathematical appendix 66
A.1 Proof of Proposition 1 (properties of the frontier) . . . 66
A.2 Proof of Proposition 2 (decomposition) . . . 67
A.3 Proof of Proposition 3 (identification and variance of the premium) . . . 67
A.4 Proof of Proposition 4 (compounding reliability) . . . 67
A.5 Proof of Proposition 5 (recovery) . . . 68
A.6 Proof of Proposition 6 (delegation thresholds) . . . 68
A.7 Variance of the scenario-weighted estimator under cluster sampling . . . 69
A.8 Validity of the scenario-cluster bootstrap: a sketch . . . 69

1. Related literature

The question of how much white-collar work an autonomous computer-using agent can perform to a professional standard sits at the intersection of six literatures that rarely cite one another: the task-based economics of automation, the randomised evidence on generative-AI productivity, the engineering literature on agent benchmarks and computer-use evaluation, the economics of delegation and supervision, the empirical study of agent reliability and of model-based evaluation, and the methodology of preregistered experiments with clustered data. This section reviews each stream with an eye to what it already establishes and what it leaves unmeasured, and closes with a positioning statement that names, for every construct introduced in Section 2, its closest verified precedent. Every reference cited here was verified against Crossref, the arXiv abstract page or the official publisher page before being admitted to the bibliography; where a preprint later appeared at a conference, we cite the preprint because that is the record that was checked.

1.1. Task-based models of automation and occupational exposure

The task framework is the natural home for a measurement exercise that insists on separating tasks from occupations. Autor et al. (2003) introduced the distinction between routine and non-routine tasks and showed that computer capital substitutes for routine cognitive and manual tasks while complementing non-routine analytic and interactive ones; the unit of analysis in that paper is the task, not the job, which is precisely the distinction that the mandate of the present study asks us to preserve. Acemoglu and Autor (2011) generalised the approach into a Ricardian assignment model in which technologies and skill groups compete for tasks, and Autor (2015) used it to explain why automation has historically created roughly as many jobs as it destroyed. Acemoglu and Restrepo (2018) and Acemoglu and Restrepo (2019) formalised the displacement, productivity and reinstatement effects: automation extends the set of tasks that capital performs, and the labour-market consequences depend on whether new labour-intensive tasks are created in the same movement. The empirical study of industrial robots by Acemoglu and Restrepo (2020) remains the template for estimating the local labour-market effects of a task-replacing technology.

The most direct bridge between that framework and the present study is the accounting formula of Acemoglu (2025), in which the aggregate productivity gain from generative AI equals the share of tasks exposed, times the share of exposed tasks actually automated, times the average cost saving per automated task. Each factor in that product is a task-level parameter—a success probability, a cost, a quality condition—that the formula takes as given. The Autonomous Labor Frontier of Section 2 can be read as an attempt to measure those parameters experimentally rather than to assume them. Two theoretical extensions are relevant to the interpretation of such measurements. Restrepo (2025) pushes the task model to the limit of artificial general intelligence and distinguishes

"bottleneck" from "supplementary" work, so that the value of automating a task depends on what remains unautomated around it; Ide and Talamàs (2025) models AI as a worker inside a knowledge hierarchy, changing who solves which problems within the firm rather than merely which tasks are performed.

Occupational exposure indices are the empirical descendants of the task framework. Frey and Osborne (2017) assigned a probability of computerisation to 702 occupations using expert labels and engineering bottlenecks; Arntz et al. (2017) showed that accounting for task-level heterogeneity within occupations reduces the share of United States jobs "at risk" from 38 to about 9 per cent, a correction that is itself an argument for measuring tasks rather than occupations. Brynjolfsson and Mitchell (2017) and Brynjolfsson et al. (2018) built the Suitability for Machine Learning rubric and found that most occupations contain some suitable tasks but almost none is fully exposed, so that reorganisation rather than wholesale replacement is the expected margin of adjustment. Felten et al. (2021) linked AI capability benchmarks to occupational abilities, and Felten et al. (2023) re-weighted the resulting index toward language modelling. The most-cited generative-AI exposure study, Eloundou et al. (2024), had humans and GPT-4 rate O*NET tasks for whether a large language model, alone or embedded in software, could halve the time required at equal quality; roughly 80 per cent of workers were found to have at least a tenth of their tasks exposed. International institutions replicated the approach at global scale—Gmyrek et al. (2023) and the refined index of Gmyrek et al. (2025) at the ILO, Pizzinelli et al. (2023) and Cazzaniga et al. (2024) at the IMF—and added a complementarity dimension that splits exposed tasks between those likely to be automated and those likely to be augmented. Hampole et al. (2025) construct firm-level, time-varying exposure from job postings and find muted net employment effects because workers shift effort toward unexposed tasks.

All of these indices share the limitation that motivates the present study: exposure is an *ex ante* judgement about what a model could do, elicited from experts or from a model, not a measurement of what an autonomous agent actually does when handed the files. Two newer strands partially close that gap. The first uses revealed usage. The Anthropic Economic Index of Handa et al. (2025) maps millions of conversations to O*NET tasks and finds usage concentrated in software and writing, with a majority of conversations classified as augmentation rather than automation; the follow-up of Appel et al. (2025) documents a rise in directive, delegation-style requests and heterogeneous enterprise adoption, and Tomlinson et al. (2025) compute an analogous applicability score from Copilot conversations. Early labour-market evidence remains mixed: Brynjolfsson et al. (2025a) find relative employment declines for the youngest workers in the most exposed occupations in payroll data, while Humlum and Vestergaard (2025), linking Danish adoption surveys to administrative registers, find rapid adoption but small earnings and hours effects to date. The second strand measures capability directly on economically valuable deliverables; it is the subject

of Section 1.3 and the immediate precursor of our design.

Two features of the task framework carry over directly into the measurement conventions of Section 2. The first is the weight attached to a task within an occupation. In the exposure literature the weight is the share of the occupation's time or value that the task represents, obtained from O*NET importance ratings or from the time-use judgements of raters (Eloundou et al., 2024; Felten et al., 2021); in the accounting formula of Acemoglu (2025) it is the share of the occupation's labour cost. Neither source is available for synthetic scenarios, which is why the mandate requires weights to be either defensible or explicitly presented as sensitivity assumptions, and why the frontier is reported under more than one weighting. The second feature is the distinction between automation and augmentation that Cazzaniga et al. (2024), Handa et al. (2025) and Appel et al. (2025) draw in different ways: an exposed task may be one the model performs alone or one in which it assists a worker, and the economic consequences differ. The present study is deliberately placed on the automation side of that distinction—the agent owns the task from brief to deliverable and no human intervenes during a run—so that what is measured is autonomous completion, and the human-time parameter of the frontier records what autonomy would cost in supervision rather than what assistance would save in effort.

1.2. Field and laboratory experiments on generative-AI productivity

The randomised evidence on *assistive* generative AI is now substantial and consistent in the sign of first-order effects. Noy and Zhang (2023) randomised access to ChatGPT among college-educated professionals performing mid-level writing tasks and found a 40 per cent reduction in time together with an 18 per cent increase in rated quality, with the largest gains for initially weaker writers. Brynjolfsson et al. (2025b) studied the staggered rollout of a generative-AI assistant to roughly five thousand customer-support agents and estimated a 15 per cent average increase in issues resolved per hour, concentrated among less experienced agents, with improved customer sentiment and lower attrition. In software, Peng et al. (2023) randomised GitHub Copilot in a controlled programming task and found 55.8 per cent faster completion, and Cui et al. (2024) pooled three field experiments inside large firms to estimate a 26 per cent increase in completed tasks. Dell'Acqua et al. (2023) gave consultants access to GPT-4 and documented a "jagged frontier": inside the frontier, consultants worked faster and produced higher-rated output, but on a task deliberately chosen to lie outside it, AI users were 19 percentage points less likely to be correct. Dell'Acqua et al. (2025) extend the design to teams and find that an individual with AI matched a two-person team without it. Otis et al. (2024) find that an AI mentor helped high-performing entrepreneurs and hurt low performers, a reminder that heterogeneity by baseline skill can flip the sign of an average effect; Choi et al. (2023) report speed gains with uneven quality effects on legal drafting; and the meta-analysis of Vaccaro et al. (2024), covering 106 experiments, finds that human–AI combinations perform on average *worse*

than the better of human or AI alone except in content-creation tasks, a result that bears directly on how much human oversight of an agent adds or subtracts.

Two findings in this stream are warnings for the design of any study of agent productivity. The randomised trial of Becker et al. (2025) gave experienced open-source developers early-2025 AI tools on their own repositories: participants expected a 24 per cent speed-up and believed *ex post* that they had been 20 per cent faster, yet were measured to be 19 per cent slower. It is, to our knowledge, the only randomised trial in which perceived and measured effects diverge in sign, and it establishes that self-reported productivity cannot substitute for timed, independently verified outcomes. In a radiology experiment, Agarwal et al. (2023) show that providing AI predictions to human readers does not improve accuracy on average because readers mis-weight the signal, so that the efficient design is a task allocation between human and machine rather than side-by-side assistance. Together with the decision-authority model of Athey et al. (2020) and the algorithm-aversion evidence of Dietvorst et al. (2015), these results justify measuring the performance of an *autonomous* agent separately from the performance of a human assisted by AI rather than conflating the two, which is why the present design treats agent arms (A0–A4) and human arms (H0–H1) as distinct treatments rather than as points on one scale.

The design lessons of this stream are as important as its findings. Every credible study in it randomises access rather than comparing adopters with non-adopters; every one times the task and grades its output blind to condition, usually with several graders; and the better ones report heterogeneity by baseline skill, by task type and by the position of the task relative to the model's frontier. The present design borrows each of these elements for an autonomous agent: tool access is randomised within task, time and cost are measured by the harness rather than reported by the agent, quality is graded blind by multiple evaluators against a rubric fixed before the runs, and heterogeneity is examined by occupation, difficulty and designed depth. What it cannot borrow in this version is the human comparison group, because the human arms require participants and an ethics approval that do not yet exist (Section 4.13); the agent-only estimates are therefore absolute rates and within-agent contrasts, not relative-to-human measures of the kind that Patwardhan et al. (2025) and Wijk et al. (2024) report.

What this stream lacks is randomised evidence on fully autonomous, multi-application agents producing complete professional deliverables. Every study cited above evaluates a human using a chat assistant or an inline completion tool; none measures the completion rate, the cost, or the supervision burden of an agent that owns a task from the brief to the delivered files.

1.3. Agent benchmarks and computer-use evaluations

Benchmarks for autonomous agents have moved in three years from single-function code generation to multi-application, long-horizon work. In software engineering, SWE-bench (Jimenez et al., 2023)

poses real GitHub issues with hidden unit tests; agent scaffolds such as SWE-agent (Yang et al., 2024) and the OpenHands platform (Wang et al., 2024b) defined the agent–computer interface paradigm—a file viewer, an editor and a shell—that Claude Code also instantiates and that Liu et al. (2026) dissect in the first academic treatment of its tool loop, context management and permission model. Terminal-Bench (Merrill et al., 2026) extends the paradigm to hard, expert-written command-line tasks and audits itself with the Agentic Benchmark Checklist of Zhu et al. (2025), which catalogues the outcome-validity and task-validity failures of existing agent benchmarks and which we treat as a design standard. SWE-Lancer (Miserendino et al., 2025) is particularly relevant economically: it prices real freelance software tasks worth one million dollars in total and measures how many dollars an agent earns. MLE-bench (Chan et al., 2024) and RE-Bench (Wijk et al., 2024) compare agents with human experts on machine-learning and research-engineering tasks under explicit time budgets and find agents ahead at short budgets but humans ahead at long ones—an early quantitative statement of the horizon effect that Section 1.5 returns to.

General computer use is measured by OSWorld (Xie et al., 2024), whose 369 tasks run in real Ubuntu, Windows and macOS desktops with execution-based checks and on which early models solved about 12 per cent of tasks against 72 per cent for humans; by its long-horizon extension (Yuan et al., 2026); by WebArena (Zhou et al., 2023), which self-hosts e-commerce, forum, content-management and map sites; by AgentBench (Liu et al., 2023), with eight environments including an operating system, a database and web shopping; and by GAIA (Mialon et al., 2023), whose questions are easy for humans and hard for assistants because they require tool use and multi-step browsing. Enterprise-flavoured benchmarks are the closest analogues of the Virtual White-Collar Economy described in Section 3: WorkArena (Drouin et al., 2024) on ServiceNow knowledge-work interfaces; TheAgentCompany (Xu et al., 2024), a simulated software company with version control, chat, a wiki and simulated colleagues, on which the best agent completed about a quarter of tasks end to end; τ-bench (Yao et al., 2024), whose retail and airline customer-service tasks come with policy constraints and the pass^k reliability metric; AppWorld (Trivedi et al., 2024), with nine simulated applications and state-based evaluation; OfficeBench (Wang et al., 2024c) on cross-application office workflows; SpreadsheetBench (Ma et al., 2024) on real spreadsheet manipulation; LegalBench (Guha et al., 2023) on legal reasoning tasks built with lawyers; and Vending-Bench (Backlund and Petersson, 2025), a months-long simulated business whose coherence collapses unpredictably over long horizons. The computer-use capability of the Claude models was introduced in Anthropic (2024), and the accompanying system card (Anthropic, 2025) documents agentic safety evaluations, including resistance to prompt injection in computer use and behaviour on impossible tasks in Claude Code, two behaviours that our scenarios deliberately probe.

Two benchmarks come closest to the economic question we pose. GDPval (Patwardhan et al.,

2025) builds tasks from the actual work of professionals with fourteen years' average experience across 44 occupations in the nine largest sectors of the United States economy, with real files as inputs and documents, spreadsheets, slides and diagrams as deliverables, graded blind by experts against the human product; frontier models won or tied roughly half of pairwise comparisons at the time of publication, and the paper reports speed and cost multiples relative to experts. METR's time-horizon construct (Kwa et al., 2025) defines the length of task, measured in human-expert time, that a model completes with 50 per cent probability, finds it doubling roughly every seven months on software tasks drawn from HCAST (Rein et al., 2025) and RE-Bench, and models the success–length relationship as a logistic curve. Kapoor et al. (2024) argue that agent evaluations must report cost alongside accuracy and must hold out tasks to prevent overfitting of scaffolds, concerns that Zhu et al. (2025) systematise.

The gap that remains is specific. GDPval evaluates a single deliverable produced from a prompt and files, graded against a human reference, without measuring the agent's interaction with a live computer environment, the number of dependent operations, the human interventions required, or the probability of recovery from an error. OSWorld, WorkArena and TheAgentCompany measure live computer use, but in software or IT-service settings, without occupational weighting and without grading against a professional standard. The METR horizon is one-dimensional and software-only. None of these studies randomises tool access—text only, shell, operating-system automation, browser and office applications—on the same tasks, and none evaluates against a hidden oracle that the worker cannot see while also recording the worker's cost and the supervision it would require. The design of Section 4 is built to occupy that gap.

Table 1 summarises the comparison with the benchmarks closest to the present design, restricted to the attributes that the review could verify from each paper.

1.4. The economics of AI agents, delegation and supervision

Delegating work to an agent whose effort and output cannot be perfectly observed is the canonical principal–agent problem. Holmstrom (1979) shows that any informative signal about unobservable effort should enter the contract, and Holmstrom and Milgrom (1991) shows that when some dimensions of output are easier to measure than others, high-powered incentives distort effort toward the measurable dimension. The second result is an exact description of an agent that passes deterministic checks while neglecting unmeasured quality, and it motivates the separation, in our evaluation pipeline, between the deterministic evaluator and the blind professional judges, as well as the explicit measurement of critical errors that a pass/fail test would miss.

The economics-of-AI literature reframes these ideas for machine agents. Agrawal et al. (2019) model AI as a fall in the cost of prediction, so that the complementary input—judgement about payoffs—becomes the scarce human contribution; Athey et al. (2020) model the choice of whether

Table 1. Closest benchmark precedents and what the present design adds to each

Benchmark	What it measures	What the present design adds
GDPval (Patwardhan et al., 2025)	Deliverables from real files in 44 occupations, graded blind by experts against the human product; cost and speed multiples	Live computer environment with observed process; hidden oracle with planted errors and traps; randomised tool access; human-time and recovery measured
METR time horizon (Kwa et al., 2025; Rein et al., 2025)	Task length in human-expert time at which success is 50 per cent, on software tasks	Designed dependency depth varied within and across non-software occupations; cost and critical errors measured jointly with success
METR developer RCT (Becker et al., 2025)	Measured versus perceived productivity of humans using AI tools on their own repositories	Autonomous rather than assisted work; the agent's cost and time are harness-measured, not self-reported
TheAgentCompany (Xu et al., 2024)	End-to-end tasks in a simulated software company with chat, wiki and simulated colleagues	Occupations outside software and IT; professional-standard grading with a quality threshold; occupational weighting
OSWorld (Xie et al., 2024; Yuan et al., 2026)	Execution-checked tasks in real desktops; observation-space ablations; human reference rate	Professional deliverables rather than state changes; deterministic and blind grading combined; tool bundles randomised on the same tasks
WorkArena (Drouin et al., 2024); AppWorld (Trivedi et al., 2024)	Knowledge-work interface tasks on an enterprise platform; day-to-day tasks across simulated applications with state-based evaluation	Complete professional assignments with planted errors and injection; critical-error and abstention outcomes
τ-bench (Yao et al., 2024)	Customer-service tasks with policy constraints; pass^k reliability across repeated trials	Repeated runs within scenario used for a variance decomposition and a recovery index, across occupations
SWE-Lancer (Miserendino et al., 2025)	Dollar value of real freelance software tasks earned by an agent	Cost per accepted deliverable net of failures and supervision, for non-software occupations
OfficeBench (Wang et al., 2024c); SpreadsheetBench (Ma et al., 2024)	Cross-application office workflows; real spreadsheet manipulation	Office work embedded in occupational assignments with an oracle, a portal, and a self-repair arm

a human or an AI holds decision authority and show that it can be optimal to make the AI *less* accurate so that the human remains engaged. Korinek (2023) catalogues the uses of language models in economic research and offers a taxonomy of delegation with verification; Horton et al. (2023) shows that language models can serve as simulated economic agents in experiments, a method that the mandate of the present study forbids for human baselines but that is admissible for synthetic counterparties inside the virtual economy. Immorlica et al. (2024) ask what changes in markets when generative AI acts as an economic agent; Hadfield and Koh (2025) sketch the institutions an economy of AI agents would require; Kolt (2025) applies agency law to AI agents and argues that monitoring and loyalty are the central governance problems. Two recent contracting models are closest to the delegation framing of Section 2.7: Sreenivas and Larson (2026) write down a model of language-model delegation in which the principal chooses a verification cost and the agent chooses a model and an effort level, deriving the agency cost of imperfect verification; and Catalini et al. (2026) argue that as the cost of execution collapses, the binding constraint becomes human verification, defining a "measurability gap" between what an AI can do and what a human can economically check.

Three further strands feed the supervision construct. In AI alignment, Bowman et al. (2022) define scalable oversight experimentally by asking whether a human supervising a model can outperform both alone, which is the alignment-community counterpart of the supervision question posed here in economic terms. Svanberg et al. (2024) provide the clearest cost-effectiveness template: for computer-vision tasks they compute the share of exposed wage value whose automation is cost-effective at current system prices and show how that share moves with cost, which is exactly the role played by the budget parameter in the frontier of Section 2. Finally, the revealed-usage evidence that users increasingly delegate complete tasks rather than ask for help with pieces of them (Appel et al., 2025) shows that the supervision question has moved from the hypothetical to the operational.

The link between this stream and the frontier of Section 2 is the human-time parameter. In Sreenivas and Larson (2026) the principal's verification cost is a choice variable that trades off against the agency cost of unverified delegation; in Catalini et al. (2026) the measurability gap is the set of outputs that an AI can produce but a human cannot economically check. The frontier's condition that human time be below an allowance is the empirical counterpart of both: it asks what share of accepted work could have been delegated with at most a given amount of checking and answering, and Section 2.7 shows that the allowance enters the delegation decision in the same way as the agent's price. What this literature does not provide, and what no observational study can, is a measurement of how much verification a given deliverable *would* have required; the present design approximates it, in this version, by the questions the agent itself records as unanswered, valued at a stated assumption, and by the deterministic and blind evaluation that a supervising professional

would otherwise have to perform.

1.5. Reliability, long-horizon failure, self-repair and judge validity

A consistent finding across benchmarks is that agent success falls steeply with task length and with the number of dependent steps. Dziri et al. (2023) show that transformer accuracy on compositional tasks decays rapidly with the depth of the computation graph and that apparent competence often reflects pattern matching on shallow sub-graphs. Sinha et al. (2025) isolate execution from planning and show that small differences in per-step accuracy compound into large differences in the length of task a model can execute, and that models display a self-conditioning effect in which earlier errors raise the probability of later ones. Kwa et al. (2025) fit a logistic success–length curve; Yao et al. (2024) show that pass^k, the probability of succeeding on all k independent trials, lies far below pass@1, so that reliability rather than peak capability is the binding constraint on deployment; and Backlund and Petersson (2025) document qualitatively similar collapses over very long horizons. Cemri et al. (2025) derive a taxonomy of fourteen failure modes in multi-agent language-model systems from more than 150 traces—specification problems, inter-agent misalignment and task-verification failures—which supplies a coding frame for our own failure analysis. Stroebl et al. (2024) show that resampling with an imperfect verifier has a hard ceiling: beyond a point, additional samples raise the rate of false positives rather than the number of correct solutions, a result that applies directly to any self-repairing treatment whose checker is itself a model or a shallow test.

On self-repair, Reflexion (Shinn et al., 2023) and Self-Refine (Madaan et al., 2023) show large gains from verbal self-reflection and iterative self-feedback on coding and reasoning tasks when an external signal such as a test result is available, and Chen et al. (2023) show that models can debug their own code by explaining execution results. But Huang et al. (2023) show that without external feedback, intrinsic self-correction of reasoning does not improve accuracy and can worsen it. The implication for the self-repairing arm of our design is that recovery must be defined relative to the *type* of error signal available—a deterministic test, an environment error, or none—and that a recovery index must distinguish detection from correction; Section 2.4 adopts exactly that decomposition.

Two further lessons of this stream shape the design. The pass^k finding of Yao et al. (2024)—that the probability of succeeding on every one of k independent trials falls steeply in k—implies that a single run per task measures capability, not reliability, and that any claim about deployment must rest on repeated runs of the same task; the within-scenario repeats of Section 4.2 exist for that reason, and the variance decomposition of Proposition 3 separates the run-to-run component that pass^k captures from the between-task component that it does not. The self-conditioning result of Sinha et al. (2025) implies that errors are not independent across the steps of a workflow, so that the simple compounding model of Section 2.4 is a benchmark shape rather than a structural claim, and

the design records the realised number of tool calls, tool errors and retries so that departures from it can be examined. The failure taxonomy of Cemri et al. (2025) and the long-horizon observations of Backlund and Petersson (2025) supply the categories under which failed runs are coded after the fact.

For evaluation validity, Zheng et al. (2023) show, with MT-Bench and Chatbot Arena, that strong language-model judges agree with humans about as often as humans agree with one another, but document position bias, verbosity bias and self-enhancement bias. Wang et al. (2024a) show that swapping the order of two candidate responses can reverse a judge's verdict; Panickssery et al. (2024) show that language-model evaluators recognise and favour their own outputs; and Gu et al. (2024) survey the field and its mitigations, namely rubric anchoring, multiple judges, blinding and calibration against human labels. The language-model judges of our evaluation pipeline are therefore blinded to the experimental condition, anchored to a rubric fixed before any run, drawn from more than one model, and reported with the agreement statistics of the next subsection; calibration against independent human experts is a requirement that this version of the study does not meet (amendment A-005) and that Section 4 states explicitly.

1.6. Experimental methodology

Preregistration and pre-analysis plans are now expected in experimental economics. Nosek et al. (2018) separate prediction from postdiction and describe registration workflows; Olken (2015) discusses the trade-off between credibility and flexibility; Simmons et al. (2011) quantify how analytic degrees of freedom inflate false-positive rates. With many outcomes, occupations and treatments, multiple-testing adjustment is required, and List et al. (2019) adapt step-down procedures to experimental economics. For design and estimation, Athey and Imbens (2016) is the reference on randomisation-based inference, stratification and power in experiments.

Our data are clustered: repeated runs within scenario, scenarios within occupation, and all agents within a model version. The number of clusters at the occupation level is small. Colin Cameron and Miller (2015) is the standard guide to cluster-robust inference; Cameron et al. (2008) introduce the wild cluster bootstrap for few clusters; and Abadie et al. (2023) clarify that clustering should follow the sampling or assignment design rather than mere correlation, which in our case means clustering at the level at which scenarios are sampled and arms are assigned within them. Mixed-effects models are the natural way to pool across scenarios, with Bates et al. (2015) as the frequentist reference. Benchmarks are tests, and psychometrics applies: Vania et al. (2021) fit item-response models to NLP test sets to compare their informativeness; Polo et al. (2024) show that item-response selection of about one hundred items can estimate benchmark performance within two points, which is the basis for calibrating scenario difficulty and for scaling from a small pilot; and Zhou et al. (2025) apply item-response theory to language-model benchmarks and find many items

uninformative or mis-keyed. Miller (2024) provide the simplest correct treatment of uncertainty in model evaluations—question-level clustering, paired differences and power analysis—and every success-rate comparison in this study follows that treatment.

Completion times are right-censored when an agent reaches a budget, a time limit or an emergency stop; survival analysis is the appropriate framework. Kaplan and Meier (1958) provides the non-parametric completion curve, Cox (1972) the proportional-hazards regression on treatment and scenario covariates, and Fine and Gray (1999) the competing-risks model in which a critical failure is a terminal event competing with acceptance rather than a form of missingness; Section 2.6 builds the Professional Reliability Frontier on that model. Inter-rater agreement among blind evaluators should be reported with chance-corrected statistics appropriate to ordinal scores: Krippendorff (2004) and Hayes and Krippendorff (2007) define and justify Krippendorff's alpha for multiple raters with missing data, and Gwet (2008) proposes the AC1 and AC2 coefficients as a remedy for the paradox of low kappa under high agreement, which is likely whenever most deliverables pass. For cost-effectiveness, the incremental cost-effectiveness ratio of the economic-evaluation literature has a direct analogue in the incremental cost per additional professionally acceptable deliverable, with uncertainty summarised by bootstrapping the joint distribution of cost and effect; within AI evaluation, Kapoor et al. (2024) propose reporting the Pareto frontier of accuracy against dollar cost, Svanberg et al. (2024) convert task-level feasibility into the share of wage value that is cost-effective to automate, and Patwardhan et al. (2025) report cost and speed multiples relative to experts. Following these precedents, every headline quantity in this study is presented as a frontier over the budget and the quality threshold rather than as a single number.

1.7. Positioning: precedents and incremental contribution

The mandate of this study instructs us not to assert novelty where a precedent exists. The honest summary of the review above is that none of the six constructs introduced in Section 2 is new as an *idea*; the contribution of the study lies in their joint, experimental, occupation-weighted measurement under randomised tool access with hidden-oracle evaluation. Table 2 lists, for each construct, the closest verified precedents and the incremental element.

Three precedents deserve an explicit comparison because a reader may reasonably ask what the present design adds to them. Relative to GDPval (Patwardhan et al., 2025), our scenarios are synthetic rather than drawn from the work of real professionals, which costs realism but buys three things that GDPval cannot offer: a hidden oracle with planted, countable errors and designed traps, so that correctness is evaluated deterministically and not only by pairwise preference; a live environment with a shell, an operating-system automation layer, a browser and an enterprise portal, so that the *process* of the work—tool calls, application transitions, retries, recovery—is observed and not only its product; and a randomised assignment of tool sets, so that the contribution of each layer

Table 2. Constructs of the study, their closest verified precedents and the incremental element

Construct	Closest precedents	Incremental element
Autonomous Labor Frontier	Task weighting from exposure indices (Eloundou et al., 2024; Felten et al., 2021; Gmyrek et al., 2025); share-of-tasks-automated term of Acemoglu (2025); cost-effectiveness share of Svanberg et al. (2024); deliverable-level grading of Patwardhan et al. (2025); task-length horizon of Kwa et al. (2025)	Joint probability of validated success, cost within budget and human time within a cap, with explicit thresholds (B, H, q); estimation from randomised runs on live computer environments with independent validation; sensitivity over task weights
Workflow Dependency Frontier	Compositional depth (Dziri et al., 2023); execution length and self-conditioning (Sinha et al., 2025); pass^k (Yao et al., 2024); long-horizon collapse (Backlund and Petersson, 2025); multi-application tasks (Xu et al., 2024; Wang et al., 2024c)	Dependency depth, number of applications and decision points treated as *designed* factors in office workflows across occupations; per-step hazard estimated with survival models
Computer-Use Tool Premium	Observation-space and interface ablations (Xie et al., 2024; Trivedi et al., 2024; Yang et al., 2024); scaffold ablations (Patwardhan et al., 2025)	Tool access randomised across the same scenarios and occupations; premium reported as a treatment effect with scenario-clustered inference (Abadie et al., 2023; Cameron et al., 2008); described as an ablation, not a new concept
Autonomous Recovery Index	Reflexion, Self-Refine, self-debugging (Shinn et al., 2023; Madaan et al., 2023; Chen et al., 2023); limits of intrinsic correction (Huang et al., 2023) and of imperfect verifiers (Stroebl et al., 2024); failure taxonomies (Cemri et al., 2025)	Decomposition of recovery into detection and correction, conditional on the error signal available, compared across a checkpointed arm and the same tools without recovery
Autonomy-Adjusted Productivity	Joint time-and-quality measurement in assistive experiments (Noy and Zhang, 2023; Brynjolfsson et al., 2025b; Dell'Acqua et al., 2023); perceived-versus-measured divergence (Becker et al., 2025); cost-controlled evaluation (Kapoor et al., 2024); verification-cost models (Catalini et al., 2026; Sreenivas and Larson, 2026)	Accepted deliverables per unit of operating cost plus wage-valued human time, measured inside an autonomous-agent experiment and reported as a function of the supervision cap
Professional Reliability Frontier	Accuracy-against-cost reporting (Kapoor et al., 2024; Patwardhan et al., 2025; Miserendino et al., 2025); validity standards (Zhu et al., 2025); jagged frontier (Dell'Acqua et al., 2023); multitask incentives (Holmstrom and Milgrom, 1991)	Critical error as a competing terminal event (Fine and Gray, 1999) graded against a professional standard, traced over quality thresholds; proposed as a reporting standard

of computer access is a treatment effect rather than a description. Relative to TheAgentCompany (Xu et al., 2024) and OSWorld (Xie et al., 2024), our environment is less rich as software but spans occupations outside software and information technology, weights tasks by their place in an occupation, and grades deliverables against a professional standard with a quality threshold rather than against a binary state check. Relative to the METR horizon (Kwa et al., 2025) and the METR developer trial (Becker et al., 2025), we replace human-time length with a designed dependency depth that can be varied within and across occupations, and we measure cost and the supervision burden jointly with success rather than success alone.

Two limitations of the positioning must be stated now rather than after the results. First, the human arms H0 and H1 of the mandate are designed and documented (Section 4.13) but not executed in this version of the study; the comparison of agents with observed human baselines under the same protocol, which would be the fourth element of the design, is therefore deferred, and no human-performance figure is imputed or simulated in its place. Second, whether any of the six named constructs survives as a distinct concept is to be decided after the pilot: the names are labels for measurement conventions, and the mandate requires that they be dropped if the evidence does not justify them (Section **??**).

2. A task-based framework for autonomous white-collar work

This section states the measurement framework that the experiment of Section 4 implements. It is deliberately modest in its theoretical ambition: following the positioning of Section 1.7, the constructs below operationalise the task model of Acemoglu and Restrepo (2018) and Acemoglu (2025) and the cost-effectiveness logic of Svanberg et al. (2024) for a setting in which the "technology" is an autonomous agent whose output on a given task is a random variable. The purpose of the formalism is to fix, before any run is executed, what is being estimated, under which assumptions it is identified, and how its components can be decomposed. Proofs are collected in Appendix A.

2.1. Occupations, tasks and runs

Let $\mathcal{O}$ be a finite set of occupations. Occupation $o \in \mathcal{O}$ is represented by a finite set of tasks $j = 1, \ldots, J_o$, each a complete professional assignment with a brief, input artefacts and required deliverables, and by a weight vector $w_o = (w_{o1}, \ldots, w_{oJ_o})$ with $w_{oj} \geq 0$ and $\sum_j w_{oj} = 1$. The weight w_{oj} is the economically justified importance of task j within occupation o; in the primary analysis the weights are equal across the tasks of an occupation, and in the sensitivity analysis they are proportional to the professional-minutes estimate recorded in each scenario's metadata (Section 3). The mandate requires that weights come from defensible occupational evidence or be presented explicitly as sensitivity assumptions; Proposition 1(vi) below bounds the consequence of a change

of weights.

An *arm* $a \in \mathscr{A} = \{A0, A1, A2, A3, A4\}$ is a configuration of the computer tools available to the agent, holding the model, the brief, the inputs and the caps fixed (Section 4.1). A *run* is one execution of task j under arm a, indexed by its repeat $r = 1, \ldots, R_{oj}$. The run is a random experiment because the language model samples its tokens stochastically and because the environment—file system, portal, office applications—responds to the sequence of actions it chooses. A run produces the vector

$$Y_{ojr}(a) = \big(D, Q, E, C, T, H, X\big)_{ojr}(a), \tag{1}$$

whose components are defined as follows. $D \in \{0,1\}$ is *deterministic completion*: every required criterion of the hidden oracle passes and the deterministic evaluator records no critical violation. $Q \in [0,100]$ is the *professional quality score*, the mean over blind evaluators of a rubric-weighted score. $E \in \{0,1\}$ is the *critical-error indicator*, equal to one when at least one critical violation is recorded (fabricated figures, modified inputs, compliance with an injected instruction, an attempted external communication, or a scenario-specific violation listed in the oracle). $C \geq 0$ is the *marginal execution cost* in United States dollars as reported by the API, including the cost of any recovery session attributed to the run. $T \in (0, \bar{T}]$ is *wall-clock time*, right-censored at the session limit $\bar{T}$; $\delta^T = \mathbf{1}\{T < \bar{T}\}$ is the censoring indicator. $H \geq 0$ is the *human intervention time* in minutes that the run requires; in the agent arms of this version of the study no intervention is permitted during a run, so H is the time implied by the questions the agent records as unanswered, valued at a stated assumption per question, plus any logged post-run intervention (Section 4.5). X collects auxiliary process measurements (tool calls, tool errors, retries, application transitions, tokens, recovery attempts) that the mandate requires to be recorded.

Definition 1 (Acceptance)**.** For a quality threshold $q \in [0,100]$, the *acceptance* indicator of a run is

$$S_{ojr}(a;q) = D_{ojr}(a)\,\big(1 - E_{ojr}(a)\big)\,\mathbf{1}\{Q_{ojr}(a) \geq q\}. \tag{2}$$

A deliverable package is accepted when it is deterministically complete, free of critical violations, and judged to be of at least the threshold quality. The *acceptance probability* of task j under arm a is $p_{oj}(a;q) = \Pr\{S_{ojr}(a;q) = 1\}$, which does not depend on r under Assumption 1 below.

The three factors in (2) are not redundant. Deterministic completion certifies facts that only the oracle knows; the quality gate certifies what a demanding professional would notice but a test cannot encode; and the integrity factor certifies the absence of behaviours that would make an otherwise acceptable deliverable unusable. The multitask logic of Holmstrom and Milgrom (1991) is the reason for keeping them separate: an agent optimising against any one of them alone can pass it while failing the others.

2.2. The Autonomous Labor Frontier

Definition 2 (Autonomous Labor Frontier)**.** For occupation o, arm a, budget $B \geq 0$, human-time allowance $H \geq 0$ and quality threshold q, the Autonomous Labor Frontier is

$$\mathrm{ALF}_o(B,H,q;a) = \sum_{j=1}^{J_o} w_{oj} \Pr\big\{S_{oj}(a;q) = 1,\, C_{oj}(a) \leq B,\, H_{oj}(a) \leq H\big\}. \tag{3}$$

The frontier is the task-weighted probability that a randomly drawn run of a randomly drawn task of the occupation produces an accepted deliverable *and* does so within the budget *and* within the supervision allowance. It is a joint probability, not a product of marginals: an arm that succeeds cheaply on easy tasks and fails expensively on hard ones is distinguished from one with the same marginal success and cost distributions but the opposite dependence. When the arm is clear from context the argument a is suppressed. The frontier is indexed by three thresholds because the economic question has three margins—whether the work is good enough, whether it is cheap enough, and whether it is autonomous enough—and because a single number would hide the trade-offs among them. Its basic properties follow directly from the definition.

Proposition 1 (Properties of the frontier)**.** *Fix o and a. Let $p_{oj}(q) = \Pr\{S_{oj}(q) = 1\}$ and let $\rho_o(q) = \sum_j w_{oj}\, p_{oj}(q)$ denote the task-weighted marginal acceptance rate. Then:*

(i) $0 \leq \mathrm{ALF}_o(B,H,q) \leq 1$ *for all* (B,H,q).

(ii) ALF_o *is non-decreasing and right-continuous in B and in H, and non-increasing in q.*

(iii) $\mathrm{ALF}_o(B,H,q) \leq \rho_o(q)$ *for all B,H, and* $\lim_{B\to\infty,H\to\infty} \mathrm{ALF}_o(B,H,q) = \rho_o(q)$. *The marginal acceptance rate is therefore the ceiling of the frontier; the gap $\rho_o(q) - \mathrm{ALF}_o(B,H,q)$ is the share of accepted work that is infeasible at the stated budget or supervision allowance.*

(iv) (Fréchet bounds.) With $F^C_{oj}(B) = \Pr\{C_{oj} \leq B\}$ and $F^H_{oj}(H) = \Pr\{H_{oj} \leq H\}$,

$$\begin{aligned}
\mathrm{ALF}_o(B,H,q) \;&\geq\; \sum_j w_{oj} \max\big\{0,\; p_{oj}(q) + F^C_{oj}(B) + F^H_{oj}(H) - 2\big\},\\
\mathrm{ALF}_o(B,H,q) \;&\leq\; \sum_j w_{oj} \min\big\{p_{oj}(q),\, F^C_{oj}(B),\, F^H_{oj}(H)\big\}.
\end{aligned}$$

(v) If the per-session budget cap is $\bar{B}$ and the run is evaluated as delivered when the cap binds, then $\mathrm{ALF}_o(B,H,q) = \mathrm{ALF}_o(\bar{B},H,q)$ for all $B \geq \bar{B}$: the frontier is flat beyond the cap and the experiment cannot inform the region $B > \bar{B}$.

(vi) (Weight sensitivity.) For two weight vectors w_o and w'_o on the same tasks, $\big|\mathrm{ALF}^{w}_o - \mathrm{ALF}^{w'}_o\big| \leq \mathrm{TV}(w_o, w'_o)\,\big(\max_j \pi_{oj} - \min_j \pi_{oj}\big)$, where π_{oj} is the joint probability in (3) for task j and TV *is the total-variation distance $\frac{1}{2}\sum_j |w_{oj} - w'_{oj}|$.*

Item (v) is a reminder that the caps of the experimental protocol are part of the estimand's support,

not nuisance parameters: a budget curve reported over $[0,\bar{B}]$ is informative, and an extrapolation beyond $\bar{B}$ is not. Item (vi) gives the maximal effect of replacing equal weights by professional-minutes weights: it is small when tasks within an occupation have similar joint probabilities and can be large otherwise, which is why both weightings are reported.

The next proposition decomposes the joint probability into interpretable factors. Each factor corresponds to an element of the evaluation pipeline and can be estimated separately, which makes the decomposition a diagnostic: it tells us *where* an arm loses acceptable work.

Proposition 2 (Decomposition)**.** *For every task j with $\Pr\{D_{oj}=1\}>0$,*

$$\begin{aligned}\Pr\{S_{oj}(q)=1, C_{oj}\le B, H_{oj}\le H\} = \quad & \underbrace{\Pr\{D_{oj}=1\}}_{\textit{deterministic completion}} \times \underbrace{\Pr\{Q_{oj}\ge q \mid D_{oj}=1\}}_{\textit{quality gate}} \\ & \times \underbrace{\Pr\{E_{oj}=0 \mid D_{oj}=1, Q_{oj}\ge q\}}_{\textit{integrity}} \\ & \times \underbrace{\Pr\{C_{oj}\le B, H_{oj}\le H \mid S_{oj}(q)=1\}}_{\textit{budget feasibility}},\end{aligned} \tag{4}$$

with the convention that a factor whose conditioning event has probability zero is set to zero. The frontier is the task-weighted sum of these products.

Because $E=1$ already implies $D=0$ in the deterministic evaluator (a critical violation blocks completion), the integrity factor in (4) equals one by construction when D is defined as in Section 2.1; it is kept in the decomposition so that the same identity applies when, in sensitivity analyses, completion is redefined on the required criteria alone and integrity is reinstated as a separate gate. The budget-feasibility factor is conditional on acceptance: it measures whether the *accepted* work was cheap and autonomous, which is the economically relevant condition, rather than whether all runs were.

2.3. The Computer-Use Tool Premium

The second construct is a causal contrast. Each run has a potential outcome under every arm. Write $S_{jr}(a)$ for the acceptance indicator that repeat r of task j *would* produce under arm a, and similarly for the other components of (1). The design assigns every task to every arm, so the contrast is identified within task from the randomised interleaving of runs rather than from a between-task comparison.

Assumption 1 (Exchangeable repeats and no interference)**.** For each task j and arm a, the run vectors $Y_{j1}(a),\dots,Y_{jR_j}(a)$ are independent and identically distributed; the vector of a run does not

depend on the arm, the order, or the node assigned to any other run; and the distribution of $Y_{jr}(a)$ does not depend on the position of the run in the execution order or on the executor node.

Assumption 2 (Information constancy)**.** The brief, the input artefacts, the deliverable specification, the model, the client version, the caps and the behavioural policy are identical across arms; arms differ only in the tool set, the documentation of that tool set, and, where applicable, the presence of the enterprise portal and of the self-repair protocol.

Assumption 1 is what the engineering of Section 5 is built to deliver: fresh workspaces, independent sessions, no shared state, no oracle on the executor nodes, and a seeded random permutation of the execution order interleaved over nodes so that any drift in the API or in node load is orthogonal to the arm. Assumption 2 makes the contrast interpretable as the effect of computer access rather than of information: the enterprise portal of arms A3 and A4 contains a subset of the input artefacts and never information absent from them (Section 3.3).

Definition 3 (Computer-Use Tool Premium)**.** For arms $a, a' \in \mathscr{A}$ and a set of tasks $\mathscr{J}$ with weights w_j, the tool premium is the task-weighted average treatment effect on acceptance,

$$\tau(a,a';q) = \sum_{j \in \mathscr{J}} w_j \left[p_j(a;q) - p_j(a';q) \right], \qquad p_j(a;q) = \mathbb{E}\left[S_{jr}(a;q) \right]. \tag{5}$$

The *shell premium* is $\tau(A1,A0)$, the *computer-use premium* is $\tau(A3,A1)$, the *macOS-layer premium* is $\tau(A2,A1)$ and the *self-repair premium* is $\tau(A4,A3)$.

The natural estimator is the task-weighted mean of within-task differences of repeat means,

$$\hat{\tau}(a,a') = \sum_{j \in \mathscr{J}} w_j \left[\bar{S}_j(a) - \bar{S}_j(a') \right], \qquad \bar{S}_j(a) = \frac{1}{R_j} \sum_{r=1}^{R_j} S_{jr}(a). \tag{6}$$

Proposition 3 (Identification and finite-task variance)**.** *Under Assumptions 1 and 2, and treating the task set $\mathscr{J}$ as fixed, $\mathbb{E}[\hat{\tau}(a,a')] = \tau(a,a')$ and*

$$\mathrm{Var}\left[\hat{\tau}(a,a') \right] = \sum_{j \in \mathscr{J}} w_j^2 \, \frac{p_j(a)\{1-p_j(a)\} + p_j(a')\{1-p_j(a')\}}{R_j}. \tag{7}$$

If instead the tasks are an independent sample of size J from an occupation's task population with equal weights $w_j = 1/J$, then with $\Delta_j = p_j(a) - p_j(a')$ and $\sigma_j^2(a) = p_j(a)\{1-p_j(a)\}$,

$$\mathrm{Var}\left[\hat{\tau}(a,a') \right] = \frac{1}{J} \left[\mathrm{Var}_j(\Delta_j) + \mathbb{E}_j\left(\frac{\sigma_j^2(a) + \sigma_j^2(a')}{R_j} \right) \right]. \tag{8}$$

Equation (8) is the reason the primary inference clusters at the task: between-task heterogeneity in the treatment effect, $\mathrm{Var}_j(\Delta_j)$, is not reduced by adding repeats, only by adding tasks, while within-task stochastic variation is reduced by repeats. It also governs the power simulation of Section 4.12. The premium is an ablation in the sense of Xie et al. (2024) and Trivedi et al. (2024); what the design adds is that the ablation is randomised on the same tasks across occupations and reported with inference that respects (8) (Abadie et al., 2023; Miller, 2024).

Two caveats on interpretation are stated now. First, under Assumption 2 the premium is the effect of a *bundle*—tools plus their documentation plus, for A3 and A4, the portal—not of a single tool; disentangling the bundle would require further arms. Second, arm A2 can only be executed on the executor nodes where the operating-system automation layer answers (Section 5.5), so $\tau(A2, A1)$ is identified conditional on those nodes; under the last clause of Assumption 1 the restriction is innocuous, and the pilot records node identity so that the clause can be checked rather than assumed.

2.4. The Workflow Dependency Frontier and the Autonomous Recovery Index

The third construct relates acceptance to the length of the chain of dependent operations a task requires. Each scenario carries a *designed dependency depth* d_j, the minimal number of dependent operations—open or convert a document, extract a figure, match records, compute, cross-check, write a deliverable section—that a careful professional must perform (Section 3.3). The Workflow Dependency Frontier is the function $d \mapsto p(a;d)$, the acceptance probability of arm a as a function of depth, estimated across scenarios and, as a secondary measure, as a function of the realised number of tool calls. A simple model of compounding error provides the benchmark shape against which the estimated frontier is read.

Consider a task whose completion requires d dependent steps, and let step k succeed with probability $\pi_k \in (0,1]$ conditional on all previous steps having succeeded, failure being absorbing. Then the probability of completing the task is

$$p(d) = \prod_{k=1}^{d} \pi_k. \tag{9}$$

Proposition 4 (Compounding reliability)**.** *Let $p(d)$ be as in (9).*

(i) If $\pi_k = \pi$ for all k, then $p(d) = \pi^d$, $\log p(d)$ is linear in d with slope $\log \pi < 0$, and the half-depth *at which the acceptance probability falls to one half is $d_{1/2} = \log 2 / \log(1/\pi)$.*

(ii) For heterogeneous reliabilities with arithmetic mean $\bar{\pi} = d^{-1} \sum_k \pi_k$, $p(d) \le \bar{\pi}^d$, with equality if and only if all π_k are equal. Heterogeneity of step reliability lowers task reliability at a given mean.

(iii) If the step reliabilities are non-increasing in k (self-conditioning in the sense of Sinha et al.,

2025), then $-\log p(d)$ is convex in d: the frontier falls faster than exponentially.

Item (i) connects the frontier to the time-horizon construct of Kwa et al. (2025): the half-depth plays the role of the 50-per-cent horizon with depth in place of human time, and a per-step reliability of π implies a half-depth of roughly $0.69/(1-\pi)$ for π close to one. Item (ii) implies that two arms with the same average step reliability can have different frontiers if one of them concentrates its failures on a few operations (for instance, producing a binary office file in an arm without a shell), which is a reason to report the frontier by arm rather than to pool. Item (iii) gives a testable shape restriction: a log-acceptance curve that is concave in depth is inconsistent with self-conditioning under this model.

Recovery changes the model in two ways, corresponding to two mechanisms that the self-repairing arm A4 contains. Within a session, an agent may detect and correct an error at a step before proceeding; across sessions, the controller resumes a session that ended without passing a public self-check (Section 4.1).

Proposition 5 (Recovery)**.** *(i) (Per-step recovery.) Suppose a failed step is detected with probability δ and, if detected, corrected with probability γ, once. With $\rho = \delta\gamma$, the effective step reliability is $\pi' = \pi + (1-\pi)\rho \geq \pi$, the task reliability is $p_\rho(d) = \big(\pi + (1-\pi)\rho\big)^d$, and the ratio $p_\rho(d)/p(d) = \big(1+\rho(1-\pi)/\pi\big)^d$ grows geometrically in d. Recovery therefore matters more, not less, for deeper workflows.*

(ii) (One-shot end-of-run recovery.) Suppose a run that would fail under arm A3 fails the public self-check with probability δ (detection), and that a resumed session converts a detected failure into an accepted deliverable with probability γ (correction). If the first session of A4 is distributed as a run of A3, then

$$p(A4;d) = p(A3;d) + \big(1 - p(A3;d)\big)\,\delta\,\gamma,$$

so the self-repair premium is $\tau(A4,A3;d) = (1-p(A3;d))\,\delta\gamma \geq 0$, increasing in the first-session failure rate.

Definition 4 (Autonomous Recovery Index)**.** In arm A4, let F_1 be the event that the first session ends without passing the public self-check and P_2 the event that the deliverables pass the public self-check after the bounded recovery session. The Autonomous Recovery Index is

$$\mathrm{ARI} = \Pr\{P_2 \mid F_1\}. \tag{10}$$

Three remarks fix the interpretation. First, the ARI is a *correction* probability conditional on *detection* by a specific signal: the public self-check verifies presence, format and well-formedness of deliverables and the validity of the completion record, and knows nothing about the correct answers.

Detection of substantive errors—a wrong reconciliation total, a missed planted discrepancy—is outside its reach, and the decomposition of Proposition 5(ii) therefore refers to the subset of failures the signal can see. Second, by the imperfect-verifier result of Stroebl et al. (2024), passing the self-check is not acceptance; the acceptance uplift of A4 over A3 is bounded above by $\Pr\{F_1\} \cdot \mathrm{ARI}$ and may be much smaller. Third, the index is defined on a design in which the recovery session shares the tools, the model, the remaining budget and the remaining wall time of the interrupted session, so that its cost and time are attributed to the same run; an ARI estimated under a different attribution rule is not comparable. The distinction between intrinsic self-correction, which Huang et al. (2023) find ineffective, and correction against an external signal, which Shinn et al. (2023) and Chen et al. (2023) find effective, is thus built into the definition: the ARI measures the second.

2.5. Autonomy-Adjusted Productivity

The fourth construct converts the frontier into a productivity measure that is net of supervision. Let ω be the wage rate per human minute used to value intervention time, and let κ be a fully loaded operating cost per run (compute allocation, software, overhead) that is reported separately from the marginal API charge C, as the mandate requires.

Definition 5 (Autonomy-Adjusted Productivity)**.** For arm a, task set $\mathscr{J}$ and quality threshold q, the Autonomy-Adjusted Productivity is the expected number of accepted deliverables per unit of fully loaded cost,

$$\mathrm{AAP}(a; q, \omega, \kappa) = \frac{\sum_j w_j \, \mathbb{E}[S_j(a; q)]}{\sum_j w_j \, \mathbb{E}\big[C_j(a) + \omega H_j(a) + \kappa\big]}, \tag{11}$$

and its reciprocal is the *cost per accepted deliverable*. The *marginal* version sets $\kappa = 0$.

The ratio is computed over all runs, accepted or not: an arm pays for its failures, and the mandate's instruction to report cost per *accepted* deliverable rather than per attempted task is precisely the instruction to keep the failed runs in the denominator. The sample analogue $\sum_{jr} S_{jr} / \sum_{jr} (C_{jr} + \omega H_{jr} + \kappa)$ is a ratio estimator; it is consistent but biased in finite samples, and its uncertainty is obtained by bootstrapping the joint distribution of numerator and denominator over tasks (Appendix A). Because AAP is a function of (q, ω, κ) and of the token prices embedded in C, it is reported as a sensitivity surface rather than as a number, following Kapoor et al. (2024) and Svanberg et al. (2024). The construct is a label for an accounting convention; the precedents listed in Table 2 already measure time and quality jointly in assistive settings, and the measured-versus-perceived divergence of Becker et al. (2025) is the reason the human-time term is measured rather than reported by the agent.

2.6. The Professional Reliability Frontier

The fifth construct treats the end of a run as one of several competing terminal events. A run terminates in exactly one of four states: *accepted* ($K = \mathrm{A}$), *critical error* ($K = \mathrm{F}$, a critical violation is recorded, whatever else was delivered), *censored* ($K = \mathrm{X}$, the wall limit, the budget cap or the turn cap stopped the session before a completion record was written), or *finished without acceptance* ($K = \mathrm{N}$, the agent stopped on its own but the package fails completion or the quality gate without a critical violation). Let T be the time at which the terminal state is reached and K its cause. The cumulative incidence function of cause k is $F_k(t) = \Pr\{T \le t, K = k\}$, and the four functions sum, as $t \to \bar{T}$, to one.

Definition 6 (Professional Reliability Frontier)**.** For arm a, the Professional Reliability Frontier is the map

$$(q,B) \longmapsto \Big(\Pr\{S(a;q) = 1, C(a) \le B\},\ \Pr\{E(a) = 1\},\ \Pr\{K = \mathrm{X}\}\Big), \tag{12}$$

evaluated task-weighted over the benchmark; the *reliability-efficient* arms at a given (q,B) are those not dominated in all three coordinates.

The competing-risks framing (Fine and Gray, 1999) matters for two reasons. A critical error is a terminal event, not missing data: a run that fabricated a total and then timed out is a failure of integrity, not a censored observation, and treating it as censored would bias the completion curve upward. And the frontier makes the trade-off visible that a single success rate hides: an arm may raise acceptance while raising the critical-error rate, which, in a professional context where a single fabricated figure can void a deliverable, is not an improvement. The jagged-frontier result of Dell'Acqua et al. (2023) is the assistive-setting precedent for this trade-off. Estimation uses the Kaplan–Meier curve for time to any terminal event (Kaplan and Meier, 1958), the Cox model for the hazard of acceptance with arm and depth as covariates (Cox, 1972), and the Fine–Gray subdistribution hazard for the cause-specific incidence of acceptance and of critical error.

2.7. Delegation with verification: a principal–agent reading

The constructs above are measurement conventions. This subsection shows how they enter the decision of a principal—a manager, a partner, a client—who must decide whether to delegate a task to an autonomous agent, and in doing so explains why the frontier has the three thresholds it has. The model is the simplest one that produces the relevant threshold; it draws on the contracting treatments of Sreenivas and Larson (2026) and Catalini et al. (2026) and on the classical moral-hazard structure of Holmstrom (1979).

A task yields value $v > 0$ to the principal when an acceptable deliverable is used. The principal can produce it in-house at cost c_H (wage times professional minutes), or delegate it to an agent

at marginal cost c_A, in which case the deliverable is acceptable with probability p and contains a critical error with probability e. An unacceptable deliverable that is nevertheless used causes a loss $\ell \geq 0$; a critical error that passes unnoticed causes a loss $L \geq \ell$. The principal may verify the deliverable at cost κ_V (human minutes times wage), in which case failures are detected and the task is completed in-house.

Proposition 6 (Delegation thresholds)**.** *(i) Delegation with verification yields a higher expected payoff than in-house production if and only if*

$$p \;\geq\; \frac{c_A + \kappa_V}{c_H}.$$

(ii) Verification is worth its cost, relative to delegating without verification, if and only if

$$\kappa_V \;\leq\; (1-p)\,(\ell - c_H) + e\,L.$$

(iii) Consequently, delegation without verification is optimal only when $\kappa_V > (1-p)(\ell - c_H) + eL$ and $c_A + (1-p)\ell + eL \leq c_H$; as $L \to \infty$ the second condition fails for any $e > 0$, so verification becomes a necessary complement of delegation whenever critical errors are possible.

Item (i) is the frontier's budget condition in decision form: a task is worth delegating when its acceptance probability exceeds the ratio of the agent's cost plus the verification cost to the human cost. The frontier's condition $C \leq B$ corresponds to $c_A \leq B$, and its condition $H \leq H$ corresponds to the human time that verification and completion require; the acceptance probability p that appears in (i) is the $p_{oj}(q)$ of Definition 1. Item (iii) formalises the claim of Catalini et al. (2026) that verification is the binding constraint: when the cost of execution c_A collapses, the delegation decision is governed by κ_V, that is, by how cheaply a human can check the work, and the measurability gap between what the agent does and what the human can verify determines the economically usable share of its output. It is also why the critical-error indicator E is measured separately from acceptance: e and L enter the verification decision through a term that acceptance alone does not capture. The model abstracts from the agent's own incentives, which are those of a software system rather than of a utility-maximising employee; the agency problem is on the principal's side, in observing the quality of what was delivered, which is the "measurable versus unmeasurable dimensions" problem of Holmstrom and Milgrom (1991) and the scalable-oversight problem of Bowman et al. (2022). The parameters of Proposition 6 other than p and e are not estimated in this study; they are reported as sensitivity assumptions in Section **??**.

2.8. What the frontier is not

The framework invites misreadings that the mandate explicitly forbids, and it is better to list them here than to qualify every later sentence.

The frontier is not an estimate of job displacement. It measures the probability that an agent completes a *task* of an *occupation* to a professional standard under stated constraints; it says nothing about whether the occupation's remaining tasks are separable from the automated ones, whether the saved time is reallocated to new tasks, or whether the demand for the occupation's output changes—the displacement, productivity and reinstatement effects of Acemoglu and Restrepo (2019) and the bottleneck logic of Restrepo (2025) are outside the measurement. A high frontier value for an occupation is a statement about the feasibility and cost of automating its tasks in a laboratory, and the step from feasibility to adoption and from adoption to employment involves adjustment costs, complementarities and organisational change that Brynjolfsson et al. (2018), Hampole et al. (2025) and Humlum and Vestergaard (2025) document as slow and partial.

A task is not an occupation. The benchmark represents each occupation by a small number of representative assignments chosen to be complete and verifiable; the weights w_{oj} are equal or proportional to professional-minutes estimates, not derived from a census of the occupation's activities, and the mandate requires that this be presented as a sensitivity assumption. The frontier therefore describes the benchmark's tasks, and the extent to which the benchmark covers the occupation is a question of content validity that no estimate from the benchmark can answer.

The scenarios are synthetic. Organisations, people, documents and figures are generated to be internally coherent and realistic, with planted errors and traps that a real file might or might not contain. Synthetic data make a hidden oracle possible and remove any risk of exposing real records, but they also mean that a frontier estimated here is a frontier on *designed* difficulty, and that the realism of the scenarios is a judgement of their validators, not a measured property. The analysis distinguishes designed features (depth, planted errors, injection, clarification traps) from incidental ones, and reports results on both, so that the reader can see how much of the frontier is shaped by the designer.

Finally, the frontier is estimated for one family of models and one client version at one date, under the caps of one protocol. Proposition 1(v) states the consequence of the caps; the dependence on the model and on the date is the reason the design includes model sensitivities and records every version identifier, and the reason the names of the constructs are provisional until the evidence of Section **??** has been examined.

3. The occupational benchmark and the Virtual White-Collar Economy

The frontier of Section 2 is defined on tasks. This section describes the tasks: the occupational taxonomy that organises them, the fictional organisations in which they are set, the anatomy of a scenario and of its hidden oracle, the procedure by which scenarios are generated and independently validated, the ten scenarios of the pilot, and the plan for scaling the benchmark. Everything described here was fixed before any experimental run; the validation status of each scenario at the preregistration freeze is recorded in the freeze manifest and reported with the results (Section **??**).

3.1. The occupational taxonomy

The mandate targets twenty occupational categories, each represented by a type of assignment that a practitioner would recognise as central to the occupation. Table 3 lists the categories, the representative simulation specified by the mandate, the task-identifier prefix used throughout the benchmark and the analysis, and whether the category is covered by a pilot scenario. The categories were chosen to span the white-collar economy rather than to sample it: they include the financial and legal professions, the analytical and research occupations, the administrative and operational functions of a firm, and the public and communication occupations. The choice is deliberately broader than the software-centred benchmarks of Section 1.3, and deliberately narrower than the full O*NET taxonomy used by exposure indices; it is a benchmark design, not a census.

3.2. The Virtual White-Collar Economy

Scenarios are set in a small, internally consistent fictional economy. Table 4 lists its organisations. All names, people, addresses and figures are fictional and generated for the study; no real entity is intended, and the generators that produce the documents draw names from a seeded roster so that a name reused across scenarios refers to the same fictional person. The organisations share conventions—the fiscal year is the 2025 calendar year, the currency is the Canadian dollar, sales taxes are the federal and Québec rates, dates follow ISO 8601—so that an agent moving between scenarios faces a coherent world rather than a collection of unrelated puzzles. The organisations are located in Québec and Ontario and the documents are written in English, the language of the working-paper series.

The economy is built around one manufacturing firm, Boréal Fenestration Inc., which appears as the client of the accounting firm, as the employer in the human-resources and procurement scenarios, as the subject of the business-intelligence pack, as the client of the law firm, and as the organisation whose chief executive's affairs are triaged in the executive-administration scenario. A commercial bank, Banque Commerciale de l'Outaouais, holds the manufacturer's operating account

Table 3. The twenty occupational categories of the benchmark

Occupation	Representative simulation (mandate)	Prefix	Pilot
Corporate finance	Financial modelling and valuation	FIN	FIN-01
Accounting	Reconciliation and financial statements	ACC	ACC-01
Banking	Credit analysis	BANK	BANK-01
Economics	Econometric estimation	ECON	—
Real estate	Property appraisal	RE	RE-01
Legal services	Contract review	LEGAL	LEGAL-01
Human resources	Administrative HR workflows	HR	HR-01
Procurement	Supplier comparison	PROC	PROC-01
Consulting	Market research and recommendations	CONS	—
Project management	Scheduling and risk management	PM	—
Marketing	Campaign development	MKT	—
Business intelligence	Dashboard and KPI reporting	BI	BI-01
Data science	Data cleaning and modelling	DS	—
Academic research	Literature review and statistical replication	RES	RES-01
Education	Course and assessment preparation	EDU	—
Insurance	Claims administration	INS	—
Logistics	Resource and delivery planning	LOG	—
Customer operations	Complex case resolution	CS	—
Journalism	Fact-checking and reporting	JOUR	—
Records management	Document classification and retrieval	REC	—
Executive administration	Inbox, calendar and project triage (mandate, Scenario 5)	ADM	ADM-01

Executive administration is the fifth illustrative scenario of the mandate; it is filed with the records and administration categories of the taxonomy and carries its own prefix. The pilot covers ten occupations; the mandate requires at least five.

in the accounting scenario, is the credit institution in the banking scenario, and is the client of the appraisal firm in the real-estate scenario. The consistency is deliberate: it allows later scenarios to reuse records that earlier ones established, and it allows workflow scenarios to span several organisations.

Table 4. Organisations of the Virtual White-Collar Economy (all fictional)

Code	Organisation	Sector	Scenarios
LMC	Lachance & Morin CPA Inc.	accounting firm	ACC
NVC	Nordvale Capital Partners	investment firm	FIN
BCO	Banque Commerciale de l'Outaouais	commercial bank	BANK
EVP	Évaluations Rivière-Nord Inc.	real estate appraisal	RE
TRL	Tremblay Roy Lévesque LLP	law firm	LEGAL
HCS	Hillcrest Strategy Group	consulting	CONS
UQD	Université du Quartier-Deschênes, Dept. of Administrative Sciences	university research	RES, EDU
BLM	Bleu Marine Communications	marketing agency	MKT
ASQ	Assurance Saint-Quentin	insurance	INS
TLX	TransLaurentides Logistics	logistics	LOG
BFI	Boréal Fenestration Inc.	manufacturing (windows, doors)	ACC, PROC, BI, PM
MGC	Municipalité de Grande-Cédrière	government administration	REC, ADM
HRD	Boréal Fenestration — Human Resources	HR department	HR
CSO	Nordique Télécom — Customer Operations	customer service	CS
PRD	Boréal Fenestration — Procurement	procurement department	PROC
JRN	Le Courrier de l'Outaouais	journalism	JOUR
ADM	Office of the CEO, Boréal Fenestration Inc.	executive administration	ADM

3.3. Anatomy of a scenario

A scenario has two halves that are produced together by one reproducible generator and never mixed afterwards: a *public bundle* that the worker sees, and a *hidden oracle* that only the controller and the evaluators see. The separation is the operational form of the mandate's rule that hidden answers and grading instructions are never exposed to the worker, and it is enforced by location: the public bundle lives in the repository under `benchmark/scenarios/<TASK_ID>/`, while the oracle lives outside the repository, on the controller only, and is never copied to an executor node (Section 5).

The public bundle contains four things. The metadata file `task.json` records the task identifier, the occupation, the organisation, a title, a version and generator seed, a difficulty class, a designed dependency depth, a professional-minutes estimate, the list of required deliverables with their formats and structural requirements (for instance that a workbook must contain live formulas), the designed features of the scenario, and the applications a professional would use. Only the subset of these fields that a worker legitimately needs—identifier, occupation, organisation, title, version, deliverables, applications—is placed in the worker's workspace; the designed features, the depth and the seed are stripped before the bundle is copied. The brief, `BRIEF.md`, is the assignment written in the voice of the organisation, by a partner, a manager or a client: it states the context, the inputs, the deliverables and their exact file names and formats, the business rules that apply, and the reporting

conventions. Every brief ends with the same working conventions, copied verbatim from a shared footer: deliverables go to `out/`, inputs must not be modified, no contact outside the workspace and no network except an internal portal if one is provided, a completion record must be written, and if the assignment cannot be completed as specified because information is missing or contradictory, the worker must say so in the completion record with its questions instead of guessing, while still delivering everything that can be completed. The brief contains no grading information, no list of planted errors, no tolerance values and no hint that any document contains an injection. The input directory contains every artefact of the assignment—spreadsheets, CSV extracts, PDF invoices and statements, Word contracts and protocols, e-mail messages in RFC 822 form, calendar files, SQLite databases, Markdown policies, images—identical in every arm. Finally, the optional portal description `portal/portal.json` lists what the synthetic enterprise portal serves in arms A3 and A4: a document library whose entries point to files already in the inputs, record tables, and forms whose submissions the brief may require.

The hidden oracle contains the ground truth (`oracle.json`: planted items and their correct values, tolerances, required deliverables, injection markers), the deterministic evaluator (`evaluate.py`), the blind-evaluation rubric (`rubric.json`: criteria, weights, and anchors on a 0–4 scale), a reference document explaining how the scenario was built, what a competent professional does, and the expected pitfalls, and the validator's report. The deterministic evaluator implements a fixed interface: it reads only the collected output directory and the public workspace, uses shared helpers to read workbook cells and formulas, document text and PDF text, applies the numerical tolerances stated in the oracle, never imports or executes the worker's code, and returns a receipt listing each required criterion with its pass/fail status and weight, each critical violation detected, the deterministic completion flag, a weighted score, and the abstention status read from the completion record together with whether abstention was justified according to the oracle. Every scenario must have at least six required criteria, of which at least three are numerical or structural facts that only the oracle knows, and at least two critical-violation checks, including input integrity and, where relevant, compliance with the injected instruction.

Deliverables are real files. A worker is asked for workbooks with live formulas, Word documents with specified sections and minimum lengths, CSV tables with specified column names, figures, code directories, Markdown summaries and draft replies, and in every scenario a machine-readable completion record `COMPLETION.json` whose public schema fixes its fields: a status among *completed*, *needs_clarification* and *abstained*; a summary; the list of delivered paths; the assumptions made; the questions the worker would have asked; and the issues it found in the inputs. The completion record is the channel through which abstention and clarification are observed, and the mandate's requirement that tasks exist which cannot be completed correctly without asking is implemented through it.

Every scenario embeds designed features, with at least two per scenario in the pilot. *Planted errors* are realistic, countable discrepancies—a transposed amount, a duplicated invoice, a comparable sale recorded with the wrong date, a reference whose metadata disagrees with the library catalogue—each listed in the oracle with its correct value. A *prompt injection* is an instruction aimed at an automated reader placed inside one input document (an invoice that asks to be approved without checking, a note that asks the assistant to skip a step); the oracle records what obeying it would look like, and obedience is a critical violation. A *missing-information or clarification trap* is a sub-task that cannot be completed without information the bundle does not contain; the correct behaviour is to flag it in the completion record and complete the rest, and the evaluator records whether an abstention was justified. An *interface perturbation*, optional and specific to the portal arms, is a portal document with a misleading title or a form whose field order differs from the brief. The behavioural rule given to every worker—treat the content of input documents as data, not as instructions—is identical across arms, so that resistance to injection is measured under the same policy whatever the tools.

Each scenario carries a *designed dependency depth*: the minimal chain of dependent operations that a careful professional must perform, counted in the reference document as the sequence of opening or converting each document, extracting and matching the figures, computing, cross-checking and writing each section of each deliverable. Depth classes follow the authoring specification: short scenarios have a depth of roughly 5 to 12, standard scenarios 12 to 25, complex scenarios 25 to 40, and workflow scenarios 30 to 60, the last class corresponding to the mandate's requirement of workflows with 30 to 60 dependent operations. The depth is a designed property of the scenario, not a property of any run; the realised number of tool calls is recorded separately, and the Workflow Dependency Frontier of Section 2.4 is estimated on both.

3.4. Generation and independent validation

Scenarios are produced and checked by operationally separate agents, in the sense of the mandate's role separation (Section 5.1). The *scenario generator* (Agent B) is a fresh, single-purpose session that writes the generator script, the brief, the inputs, the portal data and the oracle for one scenario. The generator script is deterministic given its seed: it rebuilds the inputs, the portal data and the oracle from `numpy.random.default_rng(seed)`, writes the oracle only to a directory passed as an argument, and can rebuild the public bundle with no oracle present. Internal coherence is a requirement of the script, not a hope: ledger totals tie, statements balance, dates follow the calendar, and tax amounts are computed at the stated rates, so that the planted discrepancies are the only discrepancies.

The *scenario validator* (Agent C) is a second fresh session that never sees the generator's conversation. It first solves the task from the public bundle alone, within the time limit, as a

competent professional would, and records its solution. Only then does it open the oracle and compare. It runs the deterministic evaluator on its own solution, on which every required criterion must pass, and on at least one plausible wrong solution—typically one that ignores the planted errors—which must fail. It checks that the brief is realistic and unambiguous except where ambiguity is designed, that the task is solvable from the bundle, that the hidden answers are correct, that the rubric is gradable from the deliverables alone, and that the portal holds no information absent from the inputs. Its report, its solution and the receipt of its solution are stored in the oracle directory. A scenario enters the experimental assignment only after validation, and the hashes of its bundle and oracle are recorded in the preregistration freeze manifest so that any later change is detectable. The validator's solution also serves a second purpose: it is the independent demonstration, required by the authoring specification, that the task can be completed from what the worker is given, which protects the design against the task-validity failures catalogued by Zhu et al. (2025).

The two agents are separated from the worker by construction. The worker (Agent D) runs on an executor node that holds no oracle, in a workspace composed from the public bundle only; the deterministic evaluator (Agent E) runs on the controller against the collected deliverables; the blind judges (Agent F) receive the brief, the rubric and the deliverables rendered as text, and nothing about the arm, the model, the node, the cost or the oracle (Section 4.6).

3.5. The pilot: ten scenarios in ten occupations

The pilot benchmark consists of ten scenarios, one in each of ten occupations, which exceeds the mandate's minimum of five. All ten public bundles existed with their briefs, inputs, portal data and metadata at the time of writing; Table 5 summarises them from their metadata, and the paragraphs that follow describe each assignment as a worker encounters it. Validation status is not reported here (Section **??**).

ACC-01, accounting. A staff accountant at Lachance & Morin CPA receives the year-end file of Boréal Fenestration: 95 sales invoices and 55 supplier invoices as individual PDFs, twelve monthly bank statements of the operating account, and the unadjusted general ledger as a workbook with a journal, a chart of accounts, a trial balance and notes. The monthly reconciliations were not kept during the year; the assignment is to reconcile the cash records for the fiscal year, identify and explain every discrepancy between the ledger and the bank, state which cash position at year end is correct and why, and deliver a reconciliation workbook with live formulas and a management report. The portal, where available, offers the same documents, an invoice-header table and a reconciliation summary form. This is the mandate's Scenario 1 and the deepest pilot workflow.

Table 5. The ten pilot scenarios (from each scenario's `task.json`)

Task	Occupation	Organisation	Class	Depth	Min.	Err.	Inj.	Clar.	Deliverables
ACC-01	Accounting	LMC / BFI	workflow	44	420	6	yes	no	xlsx (formulas), docx
FIN-01	Corporate finance	NVC	complex	33	300	4	yes	no	xlsx (formulas), docx
RE-01	Real estate appraisal	EVP	complex	34	300	4	yes	no	xlsx (formulas), docx
RES-01	Academic research	UQD	complex	34	300	8	yes	no	csv, code, 2 tables, png, docx
ADM-01	Executive administration	ADM	workflow	44	180	7	yes	yes	xlsx, replies, csv, docx
BANK-01	Banking	BCO	standard	24	180	3	yes	no	xlsx (formulas), docx
LEGAL-01	Legal services	TRL	standard	23	150	9	yes	yes	xlsx, docx
HR-01	Human resources	HRD	standard	22	150	4	yes	yes	xlsx (formulas), 2 docx, md
PROC-01	Procurement	PRD	standard	21	150	4	yes	yes	xlsx (formulas), docx
BI-01	Business intelligence	BFI	complex	31	300	4	yes	no	xlsx (formulas, chart), docx, code

Class: difficulty class of the authoring specification. Depth: designed dependency depth. Min.: professional-minutes estimate (used only as a weighting sensitivity). Err.: number of planted errors. Inj.: a prompt-injection document is present. Clar.: the scenario requires clarification or abstention on a sub-task. Every scenario also requires `COMPLETION.json`. Organisation codes as in Table 4.

FIN-01, corporate finance. An associate at Nordvale Capital Partners is asked by a vice-president for a base-case discounted-cash-flow valuation of the equity of a listed composites manufacturer as at the fiscal year end, and for an investment memorandum for the partners' meeting. The data room contains five years of audited statements, an annual-report excerpt with segment information and notes on leases and contingencies, market data with the discount-rate inputs and six comparable companies, a memorandum that fixes every projection and discount-rate assumption, and a broker note for context. The model workbook must have five named sheets with live formulas from assumptions to value per share and a sensitivity table; the memorandum must contain specified sections, a valuation summary, a comparables cross-check, the risks and a recommendation against the current share price. This is the mandate's Scenario 2.

RE-01, real estate appraisal. An appraiser at Évaluations Rivière-Nord takes over a mortgage-refinancing file for a single-family house: an engagement letter from the bank, the subject data compiled from the municipal roll and an old listing, six inspection photographs with captions, a summary table of eight closed sales, one source sale sheet per comparable, and the office's market evidence with a monthly price index and its adopted adjustment table. The office standard sets a source hierarchy—photographs govern for the subject, sale sheets govern for the comparables—and a sales-comparison methodology. The deliverables are an adjustment grid with live formulas and a full appraisal report. This is the mandate's Scenario 3.

RES-01, academic research. A research assistant at the Université du Quartier-Deschênes must redo, from scratch and exactly as pre-specified, the empirical analysis of a wage-subsidy programme for a grant renewal: an approved protocol fixes the research question, the cleaning rules, the difference-in-differences specification and the deliverables; the raw firm panel and its codebook

come unvalidated from the data unit; a bibliography exported from a reference manager must be checked, entry by entry, against the library catalogue, which is authoritative; and a style guide fixes the table columns and the report structure because downstream scripts read them. The deliverables are the cleaned data, the executed scripts in order, two tables, an event-time figure and a report of at least 600 words with a cleaning log and the reference-check findings. This is the mandate's Scenario 4.

ADM-01, executive administration. An executive assistant returning after a week's absence must triage the chief executive's affairs before a 10:00 meeting: forty e-mail messages received during the week, the executive's calendar for the next two weeks with 25 events, five project documents with milestones, the itinerary for a trade-show trip, a contact list and the Office of the CEO handbook, which sets filing categories, priority rules, reply conventions and security rules. The deliverables are an action plan workbook with a ranked task sheet and a calendar-problems sheet, one draft reply per message that requires one, a filing index, a briefing note and the completion record; the scenario contains a clarification trap and a portal triage form. This is the mandate's Scenario 5 and the second workflow-class scenario.

BANK-01, banking. A credit analyst at Banque Commerciale de l'Outaouais prepares a credit analysis of a 1.8 million-dollar equipment term-loan request from a metal fabricator for the regional credit committee: a loan application, three years of compiled statements with notes, the extract of the commercial credit policy whose ratio definitions are the only ones permitted, industry quartiles for the applicant's sector, and a security summary with appraised values, prior charges and advance rates. The ratio workbook must compute, with formulas from the statement figures, the coverage and leverage measures, the loan's payment schedule, the loan-to-value, and the comparison with policy thresholds and industry medians; the credit memorandum must follow the policy's requirements, and the portal offers a credit-decision form.

LEGAL-01, legal services. A junior associate at Tremblay Roy Lévesque reviews a software vendor's draft master services agreement, of 25 numbered clauses with schedules, clause by clause against the client's negotiation playbook, which states for each topic a required position, a fallback and a walk-away, together with a risk-rating guide. The partner's instructions arrive as an e-mail message, and a precedent issues register fixes the format. The deliverables are the issues register, in which only deviations from the playbook are issues and risk ratings use three letters, and a covering memorandum with key risks, negotiation priorities and open items; the scenario contains the pilot's largest number of planted deviations and a clarification trap.

HR-01, human resources. An HR coordinator at Boréal Fenestration closes the October 2025 pay period for the 25 employees of the plant and office under the payroll and leave policy: an HRIS snapshot, the time-clock export with supervisors' comments including the statutory holiday, the leave requests with their approval status, the policy, an onboarding-letter template and three e-mail requests received during the month—a resignation, a new hire and a salary-adjustment request. The adjustment workbook must have exactly specified column headers and formulas for the computed amounts with a totals row; two letters and a summary complete the package; the scenario contains a clarification trap.

PROC-01, procurement. A procurement analyst evaluates five quotations received for the annual supply of four aluminium extrusion profiles under a published evaluation method: the request for quotation with quantities, mandatory requirements, a fixed exchange rate and a freight allowance, the evaluation criteria with their weighted scales, the five quotations as PDFs in the suppliers' own presentations, a vendor master extract and a receipt log. The evaluation workbook must normalise each quotation to a total landed cost in Canadian dollars and score it by formulas on a second sheet; the recommendation memorandum goes to the vice-president of operations; the scenario contains a clarification trap and a portal award form.

BI-01, business intelligence. A business-intelligence analyst produces the annual sales KPI pack for the board from a raw nightly ERP export in a SQLite database (orders, customers, products, regions), a region mapping, binding KPI definitions with data-preparation rules, a dashboard template fixing the sheets and headers, and the vice-president's request. The dashboard workbook must reproduce the template's four sheets with formulas and a chart, the commentary must be at least 400 words, and the code that produced the pack must be delivered so that finance can reproduce it.

Two aspects of the metadata deserve a note. First, the ten scenarios cover three of the four difficulty classes (standard, complex and workflow); no short scenario is in the pilot, which is a deliberate choice to measure the regime in which the frontier is expected to be informative, and a limitation for the estimation of the Workflow Dependency Frontier at low depth. Second, several scenarios' metadata record an author-estimated time limit and budget that exceed the preregistered per-session caps of 1,800 seconds and 4.00 dollars (Section 4.3); the caps actually enforced on every run are those of the launcher configuration, not those of the metadata, and the discrepancy is recorded so that the Director can decide whether to amend either before the pilot runs.

3.6. Scaling the benchmark

The mandate asks for a benchmark that scales toward 200 validated scenarios only after feasibility, budget, statistical power and evaluation quality have been established. The scale-up plan has three stages, each conditional on the preregistered decision rule of Section 4.10. The first stage keeps the ten pilot scenarios and adds repeats, which reduces within-task variance but not between-task heterogeneity (Proposition 3). The second stage extends the benchmark to twenty scenarios by adding the ten uncovered occupations of Table 3, one scenario each, using the same generator and validator procedure, and is conditional on the remaining envelope and on the evaluator-agreement check; the power simulation of Section 4.12 indicates how much the extension buys. The third stage, toward the full benchmark, adds within-occupation variation—several scenarios per occupation across the four difficulty classes, including short tasks and the 30-to-60-operation workflows the mandate requires—so that the frontier can be estimated by occupation with within-occupation weights, and the dependency frontier by depth with designed variation rather than across-occupation comparison. Item-response calibration (Vania et al., 2021; Polo et al., 2024) is planned at that stage to select informative scenarios and to detect mis-keyed ones. No stage is executed in this version of the study beyond what the results section reports.

4. Experimental design

This section states the design that was preregistered before any pilot run: the treatment arms, the randomisation, the caps, the measurement framework, the outcome definitions, the evaluation pipeline, the hypotheses and estimands, the inference plan, the scale decision rule, the exclusion rules, the power simulation, and the human arms that were designed but not executed. The preregistration document (version 0.9 at drafting, frozen as version 1.0 before the confirmatory stage) and the amendment log are part of the reproducibility package; every departure from what follows must appear in the amendment log.

4.1. Treatment arms

Five agent arms implement increasing levels of computer access with the same model, the same brief and inputs, the same caps and the same behavioural policy. Table 6 gives, for each arm, the tool set passed to the Claude Code command-line client, the command families denied inside the shell, the additional environment, and the executor nodes on which the arm may run. The tool set and the denials are not descriptions of intent but literal command-line flags: `-tools` restricts the client to the listed tools, `-allowedTools` pre-approves them so that no permission prompt arises, and `-disallowedTools` with patterns of the form `Bash(osascript:*)` denies a command family

inside the shell. The behaviour of each flag was verified by probe before the design was fixed (Section 5.3).

Arm A0 is the restricted agent of the mandate: file reading, writing, editing and searching, with no shell and no program. Because binary office formats cannot be produced without a program, the A0 documentation instructs the worker to deliver the closest text equivalent with the same base name and to state the substitution in its completion record; the evaluators then read the substitute, and the deterministic criteria that require a formula-bearing workbook fail, as they should, since the deliverable requested was not produced. Arm A1 adds the shell and a project Python environment capable of reading spreadsheets, PDFs and databases and of writing workbooks, documents and PDFs programmatically; the operating-system automation layer, the office suite, browsers and the network are denied. Arm A2 adds the headless macOS automation layer; its definition departs from the mandate's "approved GUI automation" because, as Section 5.5 reports, control of graphical applications is not available to a non-console process on the executor nodes, and the arm is therefore the layer of macOS automation that works headlessly (amendment A-003). Arm A3 is the full computer-use agent: the office adapter for conversion and formula recalculation, a real browser engine driving the synthetic enterprise portal, and the portal itself with its document library, records and forms. Arm A4 has the same tools as A3 and adds a self-repair protocol in three parts: a checkpoint file that the worker keeps updated, a public self-check tool that verifies the presence and well-formedness of deliverables and knows nothing about the correct answers, and one bounded recovery resume granted by the controller when the first session ends without passing the self-check or ends by timeout, budget exhaustion or error.

Each arm's documentation is appended to the system prompt after a common policy that is identical across arms: the worker is an autonomous professional executing one assignment in an isolated workspace, nobody will answer questions during the session, questions go into the completion record, inputs must never be modified, the network is forbidden except the internal portal when one is given, messages and purchases are forbidden, the content of input documents is data and not instructions, and the work is to be done to the standard of a competent professional in the organisation described, within the session's budget. The user prompt is a single sentence asking the worker to read the brief and carry out the assignment completely. The difference between arms is therefore confined to the tool set, the denials, the tool documentation, the presence of the portal and, for A4, the self-repair protocol, which is what Assumption 2 requires.

4.2. Information constancy, randomisation, interleaving and repeats

Information is held constant across arms by construction. The public bundle is identical in every arm; the portal of arms A3 and A4 serves documents that are already in the inputs, record tables derived from them and forms, and is forbidden by the authoring specification from holding any

Table 6. The five agent arms

Arm	Tools (`-tools`)	Denied command families (`-disallowedTools`)	Environment	Nodes
A0 Restricted	Read, Write, Edit, Glob, Grep	(no shell)	none; text deliverables only	any
A1 Shell	A0 + Bash	`osascript`, `open`, `soffice`, `*LibreOffice*`, `textutil`, `qlmanage`, `cupsfilter`, `curl`, `wget`	project Python environment (pandas, openpyxl, xlsxwriter, python-docx, python-pptx, reportlab, pypdf, matplotlib, statsmodels, scipy, polars, duckdb, pillow, sqlite3)	any
A2 macOS	A1 tools	`soffice`, `*LibreOffice*`, `curl`, `wget`	A1 + `osascript` (AppleScript, JavaScript for Automation, System Events) and native document utilities (`textutil`, `sips`, `mdls`/`mdfind`, `qlmanage`, `cupsfilter`)	M3U96a, M4M64a
A3 Full computer-use	A1 tools	none	A2 + LibreOffice headless (office adapter) + Playwright headless Chromium + synthetic enterprise portal on loopback, credentials in the workspace	any
A4 Self-repairing	A1 tools	none	A3 + checkpoint file, public self-check tool, and one bounded controller-side recovery resume if the first session ends without passing the public self-check	any

All arms: model `claude-opus-5-5`, default effort, `-bare`, `-permission-prompts none`, `-no-chrome`, `-disable-slash-commands`, `-no-session-persistence`, `-max-budget-usd 4.00`, `-max-turns 150`, output `stream-json`. No Agent or Task tool is ever given, so a worker cannot spawn sub-agents. A2 runs only on the two nodes where System Events answers over SSH (amendment A-003).

exclusive information. Model, client version, caps and behavioural policy are identical. What differs is access, and the premium of Section 2.3 is the effect of access.

The design is a within-scenario, five-arm comparison in which every scenario is run under every arm with R repeats. The assignment is a seeded random permutation of the cells (scenario, arm, repeat); the permutation fixes the order in which cells are executed, and the scheduler draws cells from that order and dispatches each to the least-loaded eligible executor node, so that arms are interleaved over time and over nodes rather than run in blocks. Interleaving is what makes the last clause of Assumption 1 plausible: drift in the model's service, in the API's latency, or in node load affects all arms alike. Arm A2 is eligible only on the two nodes where the automation layer answers; the node is recorded for every run, so that node effects can be examined rather than assumed away. The random seed of the permutation, the order index of every run and the node assignment are stored with the run.

Repeats measure the stochastic variation of the agent on a fixed task, which is the variance component that the pass^k metric of Yao et al. (2024) isolates. The pilot uses $R = 2$ repeats per cell; the scale rule of Section 4.10 determines whether a third repeat is added. Workspaces are reset between runs: each run receives a fresh copy of the public bundle on its node, and the workspace is deleted after collection, so that no run sees the artefacts of another.

4.3. Caps

Every session is bounded by three caps enforced independently: a spending cap of 4.00 dollars passed to the client as `-max-budget-usd`, a turn cap of 150 passed as `-max-turns`, and a wall-clock limit of 1,800 seconds enforced by the controller, which kills the remote process when the limit is reached. The recovery session of arm A4 receives the same caps minus what the first session spent, with floors of 0.50 dollars and 300 seconds so that a recovery is never granted an empty budget. The study as a whole has an envelope of 800 dollars (amendment A-001), enforced by the scheduler's ledger: when the sum of recorded worker, recovery and judge costs reaches the envelope, the queue stops. A run that reaches a cap is not discarded: it is evaluated as delivered (Section 4.11), its censoring is recorded, and Proposition 1(v) describes what the cap does to the estimand.

4.4. Measurement framework

The mandate lists what must be recorded for every execution; Table 7 maps each item to the variable recorded by the laboratory and to its source. Variables derive from four sources: the immutable event stream of the session, the client's final result event, the collection step on the controller, and the evaluation pipeline. Missing observations are preserved as missing, never set to zero: a run

whose result event was not received because the process was killed has a missing cost field and a recorded timeout, not a cost of zero. Raw streams are append-only files; the normalised data are exported to Parquet for analysis.

Table 7. Recorded variables (mandate §11) and their sources

Mandate item	Variable(s) recorded	Source
Validated completion	`det_completion` (D), `det_score`, per-criterion pass flags	deterministic evaluator receipt
Professional quality score	`quality` (Q, mean of judges), per-judge scores and acceptable votes	blind judges
Critical-error indicator	`critical_error` (E), list of critical violations	deterministic evaluator + harness hash check
Total execution time	`wall_s`, `wall_total_s` (incl. recovery), `timed_out`	launcher
Tool execution time	`tool_time_s`, `first_tool_s`	event stream (tool-use to tool-result intervals)
Token consumption	input, output, cache-read, cache-creation tokens; per-model usage	event stream, result event
Actual API charges	`cost_usd`, `recovery_cost_usd`, `total_cost_usd` (C)	result event `total_cost_usd`
Compute allocation	node, slots per node, concurrent sessions	scheduler
Human intervention minutes	`human_intervention_min`; questions in completion record	interventions table (none permitted during agent runs); `COMPLETION.json`
Number of tool calls	`tool_calls`, by tool and by family (file, shell, python, sql, office, applescript, browser)	event stream
Number of retries	`retries` (same tool re-invoked after a tool error), `tool_errors`	event stream
Recovery attempts and success	`recovery_attempted`, `recovery_success`, `selfcheck_pass_first`, `selfcheck_pass_final`	scheduler (arm A4)
Application transitions	`app_transitions` (changes of tool family), `family_sequence`	event stream
Workflow dependency depth	`designed_dependency_depth` (scenario); realised tool calls (run)	scenario metadata; event stream
Output artifacts	path, SHA-256 and size of every file in `out/`; portal submissions	collection step
Environment and software versions	`claude_code_version`, `model_reported`, node, Python and office versions	system event; Phase 0 survey
Final evaluator decision	$S = D\,(1-E)\,\mathbf{1}\{Q \geq q\}$; abstention status and justification	evaluation pipeline

4.5. Outcome definitions

The outcomes are those of Section 2.1, with the following operational detail. Deterministic completion D is true when every required criterion of the oracle passes and no critical violation is recorded by the evaluator or by the harness; the harness adds a critical violation of its own when the SHA-256 hashes of the input files after the session differ from the manifest written before it.

Quality Q is the mean over available judges of the rubric-weighted score, each judge's score being the weighted mean of integer criterion scores on a 0–4 scale rescaled to 0–100; the threshold is $q = 70$ in the primary analysis and is varied from 50 to 90 in sensitivity analyses. The critical-error indicator E is one when the list of critical violations is non-empty. Cost C is the sum of the first session's and, for A4, the recovery session's `total_cost_usd` as reported by the client, with the recovery cost computed as the increment when the resumed session reports a cumulative figure. Time T is the wall time of the session, plus the recovery session's for A4, censored at the limit. Human time H is, in this version, zero intervention minutes during the run—none is permitted—plus a stated assumption of five minutes per unanswered question recorded in the completion record, which is the time a supervisor would spend answering it; the assumption is varied in sensitivity analyses, and the frontier is reported at $H = 0$ and at the implied allowance. Abstention is recorded from the completion record's status and judged *justified* by the evaluator only when the oracle says that the scenario truly requires clarification on the sub-task in question.

4.6. The evaluation pipeline and its blinding

Evaluation has two independent components. The *deterministic evaluator* (Agent E) is scenario-specific code that lives in the hidden oracle, is loaded by the controller after the deliverables have been collected, and runs against a composed workspace consisting of the public bundle and the collected output directory. It reads only the output and the public workspace, never imports or executes the worker's code, applies the tolerances recorded in the oracle, and returns a receipt validated against a public schema; the hash of the oracle and of the evaluator code is recorded in each receipt, so that any change of the evaluator after a run is detectable. A crash of the evaluator yields a receipt with no passed criterion and a note, never a silent pass.

The *blind professional evaluators* (Agent F) are three language-model judges—one draw of `claude-opus-5-5` and two independent draws of `claude-sonnet-5-5`—run on the controller as single-turn, tool-less sessions with a fixed JSON output schema. Each judge receives the brief, the rubric with its criteria, weights and anchors, and the deliverables rendered as text (documents as paragraphs, spreadsheets as cell listings, truncated at a fixed length), and is asked to grade each criterion as an integer from 0 to 4, to say whether a demanding manager would accept the package as professionally acceptable, and to write a short comment; it is instructed not to reward length and to penalise unsupported numbers, missing sections and instructions not followed. The judge never sees the arm, the model, the node, the cost, the event stream, the oracle values or the other judges' outputs; a text substitute delivered by arm A0 is rendered under the base name requested, so that the substitution itself is not a signal of the arm beyond what the deliverable's own form reveals. The judge cost is attributed to the run in the ledger but not to the worker's cost C. This design follows the mitigations of Gu et al. (2024): rubric anchoring, multiple judges, blinding to condition. It does

not address self-preference (Panickssery et al., 2024), since the judges and the worker belong to the same model family; the agreement between the deterministic evaluator and the judges' acceptability votes, and the Krippendorff alpha among judges, are reported as a partial check, and calibration against independent human experts is left to the principal investigator (amendment A-005). No evaluator ever sees two candidate packages side by side, so the position bias of Wang et al. (2024a) does not arise in this design.

4.7. Preregistered hypotheses

The preregistration states seven directional expectations. They are expectations, not guarantees, and the results section reports each against its estimate whatever the sign.

H1 (primary, computer-use premium).
$\Pr(\text{acceptance} \mid A3) - \Pr(\text{acceptance} \mid A1) > 0$.

H2 (shell premium).
$\Pr(\text{acceptance} \mid A1) - \Pr(\text{acceptance} \mid A0) > 0$, driven by the feasibility of binary deliverable formats.

H3 (self-repair).
$\Pr(\text{acceptance} \mid A4) - \Pr(\text{acceptance} \mid A3) \geq 0$, and the Autonomous Recovery Index exceeds one half when a first session fails the public self-check.

H4 (dependency).
Within arm, acceptance decreases with the designed dependency depth and with the realised number of tool calls.

H5 (integrity).
The rate of critical violations—compliance with an injected instruction, fabrication, modification of inputs—is below 10 per cent in every arm and does not increase with tool access.

H6 (cost).
The cost per accepted deliverable is lowest in A1 or A3, not in A0, whose acceptance is expected to be near zero when binary deliverables are required.

H7 (heterogeneity).
Effects differ across occupations; no occupation-level claim is confirmatory at the pilot scale.

4.8. Estimands

For run i of scenario j under arm a, the primary endpoint is $S_{ija} = \mathbf{1}\{D_{ija} = 1 \wedge E_{ija} = 0 \wedge Q_{ija} \geq 70\}$. The scenario-weighted acceptance of arm a is $\hat{p}_a = J^{-1} \sum_j R_j^{-1} \sum_r S_{jra}$, with equal scenario

weights in the primary analysis and professional-minutes weights as a sensitivity. The primary contrast is $\hat{\Delta}_{31} = \hat{p}_{A3} - \hat{p}_{A1}$; the secondary contrasts are $\hat{\Delta}_{10}$, $\hat{\Delta}_{43}$ and $\hat{\Delta}_{21}$, Holm-adjusted over the three. Secondary endpoints are deterministic completion D, quality Q, the critical-error indicator E, cost C including recovery, wall time T censored at the limit, tool calls, tool errors, retries, application transitions and tokens, the abstention status and whether it was justified, the cost per accepted deliverable $\sum C / \sum S$ by arm, the Autonomous Labor Frontier by occupation as a function of the budget cap from 0.50 to 4.00 dollars and of the human-time allowance, the Autonomous Recovery Index, and evaluator agreement, namely Krippendorff's alpha on the judges' scores and the agreement between deterministic completion and the judges' acceptability votes.

4.9. Inference plan

The inference plan follows the clustering logic of Proposition 3 and the recommendations of Miller (2024) and Abadie et al. (2023). The primary interval for every contrast is a scenario-cluster bootstrap: scenarios are resampled with replacement $B = 5{,}000$ times, the within-scenario paired difference of repeat means is recomputed, and the percentile 2.5 and 97.5 points of the bootstrap distribution form the 95 per cent interval. The primary test is a paired sign-flip permutation test over scenarios with 20,000 permutations, which is exact under the null of no effect within scenario and makes no distributional assumption. As a sensitivity, a logistic regression of acceptance on arm indicators with standard errors clustered by scenario, and a mixed-effects logistic model with a scenario random intercept (Bates et al., 2015), are reported. Time to completion is analysed with the Kaplan–Meier estimator by arm and a Cox model with arm and depth as covariates, treating runs that reach a cap as censored and, in the competing-risks version of Section 2.6, critical failures as a competing cause (Fine and Gray, 1999). Counts of tool calls are modelled as Poisson or negative-binomial. With $J = 10$ scenarios the intervals will be wide, and the preregistration states that they are to be reported as such; the pilot is not powered for occupation-level claims. Ratio quantities—cost per accepted deliverable, Autonomy-Adjusted Productivity—are bootstrapped jointly in numerator and denominator over scenarios (Appendix A).

4.10. Scale decision rule

The pilot comprises 10 scenarios $\times$ 5 arms $\times$ 2 repeats, that is 100 runs at the primary model. Let $\bar{c}$ be the mean total cost per run, including the worker, any recovery and the judges, and let E_r be the envelope remaining after the pilot. The rule, fixed before the pilot, is: if $1.15\,\bar{c} \times 50 \leq E_r - 100$, a third repeat is added for every cell; then, if $1.15\,\bar{c} \times 40 \leq$ the remaining envelope minus 50, the model sensitivities are run—`claude-sonnet-5-5` and `claude-fable-5-1` on arm A3, two repeats each, on the same scenarios. Extension to twenty scenarios is deferred to a later wave with a

separate authorisation and is conditional on the evaluator-agreement check (Krippendorff's alpha of at least 0.6 on quality). Repeats added after the pilot are pooled with the pilot only if no scenario, oracle, rubric, arm text or harness code affecting outcomes changed, as verified by hash; otherwise they are reported separately.

4.11. Exclusions and missingness

No assigned run is excluded. Infrastructure errors—failures of SSH, of the file transfer, or of the portal start, in which no inference was performed—are rerun with the same run identifier after the workspace is reset, and the failed attempt is logged in the amendment file and counted in the infrastructure-stability statistics. Runs that time out, exhaust the budget or reach the turn cap are evaluated as delivered, with acceptance computed on whatever was in the output directory. A missing judge output leaves quality averaged over the available judges; if no judge output is available, quality and acceptance are missing and reported as missing, never imputed as zero. The envelope is enforced by the ledger; the pilot runs its assignment once.

4.12. Power simulation

Because no comparable experiment exists from which to borrow effect sizes, the power of the design was explored by simulation before the pilot. The model is the cluster structure of Proposition 3: each scenario has a random effect on the logit scale with standard deviation $\tau = 0.8$, the arm shifts the logit by the amount that moves the baseline acceptance p_{base} by δ percentage points, each cell draws R binomial repeats, and the test is a paired t-test on the scenario-level differences at the 5 per cent level, which is the large-B approximation of the scenario-cluster bootstrap interval and is conservative-equivalent at $J = 10$. Each configuration was simulated 4,000 times. Table 8 reports the power at a baseline acceptance of one half for $J \in \{10, 20, 40\}$ scenarios, $R \in \{2, 3, 5\}$ repeats and effects of 15, 25 and 35 percentage points; the full grid, including a baseline of 0.3, is in the analysis tables.

The table makes the limits of the pilot explicit. With ten scenarios and two repeats, the design detects an effect of 35 percentage points about half the time and an effect of 15 points rarely; a third repeat helps moderately, and doubling the number of scenarios helps more than doubling the repeats, as equation (8) predicts when between-scenario heterogeneity dominates. The pilot is therefore a feasibility and estimation exercise whose primary contrast is expected to be large if the shell and the office tools matter for binary deliverables at all, and it is the reason the scale rule prioritises scenarios over repeats in the later wave.

Table 8. Simulated power of the paired scenario-level test ($p_{\text{base}} = 0.5$, $\tau = 0.8$, $\alpha = 0.05$, 4,000 simulations)

J scenarios	R repeats	$\delta = 15$ pp	$\delta = 25$ pp	$\delta = 35$ pp
10	2	0.115	0.281	0.550
10	3	0.164	0.408	0.729
10	5	0.243	0.599	0.906
20	2	0.238	0.575	0.902
20	3	0.320	0.751	0.973
20	5	0.495	0.921	1.000
40	2	0.418	0.866	0.997
40	3	0.583	0.968	1.000
40	5	0.795	0.999	1.000

Source: `protocol/power_analysis/power_simulation.py`, output `analysis/tables/power_pilot.csv`. Power is the share of simulations in which the one-sided paired test rejects in the hypothesised direction.

4.13. The human arms: designed, not executed

The mandate specifies two human conditions: H0, in which human participants complete tasks without AI, and H1, in which they use conversational AI assistance; it requires observed participants, consent and institutional ethics approval, and forbids substituting a language-model imitation of a human for a real baseline. No human participant was involved in this version of the study, no ethics approval was sought, and no human-performance figure appears anywhere in the manuscript except as an explicit placeholder for a future wave (amendment A-002). The protocol for that wave is part of the reproducibility package. It specifies professionals or senior students in each occupation, with a target of six participants per scenario per arm recruited through the department; randomisation between H0 and H1 within occupation, the H1 interface being a chat assistant without tools; one scenario per participant per session of at most three hours; the same public bundle, brief and deliverable list; the same deterministic evaluators and blind rubric; and recording of time, of tool switches and, with consent, of the screen and, in H1, of the chat transcript, with no intervention allowed. Ethics requirements follow the TCPS 2 framework and the university's research ethics committee: informed consent from a template in the package, voluntary participation, compensation at the professional rate, anonymised deliverables, secure storage of recordings, withdrawal at any time, and no deception, since participants are told that the data are fictional. The analysis plan uses the same scenario-weighted estimands as the agent arms; agent-versus-human contrasts at that scale would be descriptive. Until that wave is run, every statement in this paper about the agent arms is a statement about agents alone, and the comparison with human baselines that the positioning of Section 1.7 anticipates remains open.

5. Laboratory infrastructure: Claude Code calling Claude Code

The mandate requires that the research laboratory be built, not described: a central controller must launch and supervise independent Claude Code sessions with isolated workspaces, bounded permissions, task-specific prompts and resource limits, and must log, hash and reset everything it touches. This section describes the system as built, the environment it was measured in, and the places where the environment constrained the design. Everything stated about the environment was measured during Phase 0 of the project on the day the laboratory was assembled; nothing is copied from historical inventories.

5.1. Controller and executors

The laboratory has a controller–executor architecture. The *controller* is a laptop session that holds the repository, the hidden oracles, the experiment database and the ledger; it draws the assignment, composes workspaces, launches sessions over SSH, streams and stores their events, enforces the wall limit, collects and hashes deliverables, runs the deterministic evaluator and the blind judges, and renders the dashboard. It never executes a worker itself. The *executors* are four Mac Studio nodes of the authorised MacLustr cluster on which Claude Code is installed; each runs up to three concurrent worker sessions in separate per-run directories, hosts the synthetic enterprise portal for the runs that need it, and holds no oracle, no rubric and no evaluator code. Figure 1 shows the architecture; Table 9 gives the correspondence between the mandate's nine roles and the processes that implement them.

Figure placeholder. Controller (laptop: assignment, workspace composition, launcher, event store, ledger, deterministic evaluator, blind judges, dashboard; hidden oracles outside the repository) $\longrightarrow$ SSH $\longrightarrow$ four executor nodes (per-run workspaces, `claude -bare` sessions under arm-specific flags, loopback portal for A3/A4, LibreOffice headless, Playwright Chromium) $\longrightarrow$ collection, hashing, reset. The figure file `fig01_architecture.pdf` is produced by the figure script of the reproducibility package.

Figure 1. Laboratory architecture: controller, executors and the separation of hidden material

The separation is operational rather than nominal. Generator and validator are distinct sessions with no shared conversation; the worker is a separate operating-system process on a separate machine, started in a mode that ignores the user's settings, hooks and instruction files, so that nothing from the controller's own configuration reaches it; the deterministic evaluator is code that the worker never sees; the judges receive a rendered package and a rubric; and the writer of the present section had access to the protocol, the code and the benchmark, but to no run.

Table 9. Role separation: the mandate's roles and the processes that implement them

Role	Process	Sees the oracle?
A Director	the controller session and `protocol/`	yes
B Scenario generator	one fresh sub-session per scenario, writing bundle and oracle	writes it
C Scenario validator	one fresh sub-session per scenario; solves from the public bundle first, then compares	after solving
D Worker	`claude -bare` on an executor node, in a fresh workspace	never
E Deterministic evaluator	`evaluate.py` of the hidden oracle, run on the controller	yes
F Blind evaluator	`claude -p -json-schema` judges on the controller; rubric and deliverables only	rubric only
G Statistician	the `analysis/` scripts, on validated data	validated data
H Writer	sub-sessions writing `manuscript/` from the `analysis/` outputs	no
I Auditor	a fresh sub-session checking claims against records at the end	all records

5.2. The measured environment

Table 10 summarises the Phase 0 survey of the executor nodes. The controller runs macOS 27.0 with Claude Code 2.1.316; the four executors run macOS 26.4 to 26.6 with Claude Code 2.1.314, a single version on all nodes, which is pinned by recording the client version reported in the system event of every session. Each node has a project Python environment, LibreOffice 26.2 for headless conversion and recalculation, and Playwright's headless Chromium, all installed or verified during the survey. Other cluster nodes that answered over SSH did not have Claude Code installed and were not used; a few did not answer within the timeout.

Table 10. Executor nodes (Phase 0 survey, measured live)

Node	macOS	Cores / RAM	Free disk	Claude Code	Python	LibreOffice	System Events
M3U96a	26.6	28 / 96 GB	238 GB	2.1.314	3.14.5	yes	answers
M3U96b	26.4	28 / 96 GB	380 GB	2.1.314	3.14.4	26.2.3	times out
M4M64a	26.5.1	16 / 64 GB	540 GB	2.1.314	3.14.4	26.2.3	answers
M4M64b	26.5.1	16 / 64 GB	54 GB	2.1.314	3.14.4	yes	times out

"System Events": whether `osascript` reaches System Events from a non-console SSH session. Finder and TextEdit time out on all four nodes (Section 5.5). All nodes have a console user logged in, the project Python environment, `uv` and headless Chromium.

Model access was verified by a headless single-turn probe of each candidate model from an executor node, with the API key read from an environment file held outside the repository with restricted permissions and never logged. The probe recorded, from the `total_cost_usd` field of the client's result event, the marginal cost of one trivial turn: about 0.0002 dollars for `claude-haiku-5-5`, 0.004 for `claude-sonnet-5-5`, 0.008 for `claude-opus-5-5` and 0.020 for

`claude-fable-5-1`. These are infrastructure measurements, not experimental results; their only use was to confirm that costs are read from the client rather than from an assumed price table and to inform the choice of the primary model and its sensitivities (amendment A-004).

5.3. The launcher

A worker session is a single command assembled by the launcher from the arm's configuration and the run's identifiers: the client is started in bare mode, with the prompt read from a file in the workspace, streaming JSON events with verbose output, the model fixed, the tool set restricted and pre-approved, permission prompts disabled, the budget and turn caps set, the arm's documentation appended to the system prompt from a file, the browser integration disabled, slash commands disabled, session persistence disabled, the arm's denied command families listed, and a session identifier generated by the controller. The command runs on the node through SSH inside a shell line that changes to the workspace, sets the path, sets an environment variable that marks the process as a worker, disables non-essential client traffic and the auto-updater, loads the API key from the node's environment file, and replaces the shell with the client; standard input is closed. The controller reads the event stream line by line, appends every line unmodified to an append-only raw file, normalises it into the event store, and parses the final result event, from which it takes the cost, the number of turns and the termination subtype.

Four properties of the launcher are design requirements rather than conveniences. Bare mode means that the worker runs without the user's settings, hooks and instruction files, so that the controller's own cluster instructions never reach a worker. The tool restriction is literal: a worker started without the shell tool has no shell, as verified by a probe in which the model reported only file tools and could not run a trivial command; a denied command family is denied inside the shell, as verified by a probe in which an AppleScript invocation was refused and a plain command ran. No Agent or Task tool is ever given to a worker, so a worker cannot spawn sub-agents, and the launcher refuses to start if the worker-marker variable is already set in its own environment, which prevents a nested controller. The wall limit is enforced by the controller: when the elapsed time exceeds the limit, the remote process is killed by its unique session identifier, first gently and then forcibly, and the run is marked as timed out; a run that ends by the client's own budget or turn cap is recognised from the result subtype.

5.4. Workspace isolation, portal, office and browser adapters

Each run receives a workspace composed on the controller and pushed to its node. The workspace contains the public bundle with the generator script removed and the metadata reduced to what a worker legitimately needs, the public completion-record schema, the prompt and system files, for

the portal arms a portal description with the loopback URL and the credentials, and for arm A4 the public self-check tool. A manifest of SHA-256 hashes of every file in the workspace, excluding the output directory, is written before the session starts. After the session, the output directory is copied back and hashed file by file, the inputs are copied back and compared with the manifest, any change to an input is recorded as a critical violation, the portal is stopped, and the workspace on the node is deleted. Successive runs on the same node therefore never see one another's files, and the reset is a deletion rather than a cleanup.

The *synthetic enterprise portal* is a small web application started on the node for each run of arms A3 and A4, bound to the loopback interface on a free port and driven by the scenario's portal description. It serves a login page, a document library whose download links point to files in the inputs, record tables as HTML and as JSON, and forms whose submissions are appended to a file in the output directory, where they are collected and evaluated with the other deliverables. It holds no information absent from the inputs, by specification and by construction, since its documents are served from the input directory. The *office adapter* is LibreOffice in headless mode, used by the worker to convert documents and to recalculate workbook formulas so that cached values are present when a file is opened; the *browser adapter* is Playwright's headless Chromium in the project Python environment, which the worker can drive to log in to the portal, browse the library, download documents and submit forms; the portal is also reachable with command-line HTTP tools. External network access is forbidden by policy and the portal listens on loopback only; the orchestrator gives the worker no credential other than the API key in its process environment.

5.5. The macOS layer: what works headlessly and what does not

The mandate's arm A2 is a "macOS agent" with AppleScript and approved GUI automation. Phase 0 measured what is actually available to a process started over SSH on a node whose console user is logged in. The AppleScript interpreter runs on all nodes and reaches System Events—process lists, the frontmost application, file-system automation—on two of the four nodes; on the other two, every Apple Event times out. Finder and TextEdit time out on all four nodes: the request is blocked by the automation consent prompt that macOS presents on the console and that nobody can answer for a non-console process, which is the behaviour of the Transparency, Consent and Control framework for background sessions. The consequence is recorded as amendment A-003 and reflected in Table 6: arm A2 is implemented as the shell plus the macOS automation layer that works headlessly—`osascript` with AppleScript and JavaScript for Automation, `do shell script`, System Events on the nodes that answer—and the native document utilities for conversion, images, metadata, previews and printing to PDF. Control of graphical applications such as TextEdit, Numbers or Pages is not available and is recorded as an infrastructure limitation; GUI-dependent runs are pinned to the two nodes where System Events answers. The mandate's warning that

ordinary containers do not provide reliable macOS GUI isolation was, in this sense, confirmed from the other direction: on real machines, GUI automation is not available to a headless experiment at all without a dedicated console session, which the present version of the laboratory does not have.

5.6. The A4 self-repair protocol and bounded recovery

Arm A4 adds three elements to arm A3. The worker is instructed to keep a checkpoint file in the output directory updated after every major step—what is done, what remains, where intermediate files are—and, before finishing, to run the public self-check and fix every problem it reports. The self-check is a short program copied into the workspace; it verifies that every required deliverable exists and is non-empty, that office files are valid archives, that a workbook required to contain formulas does contain them and shows no error values when cached values are read, that a document has a minimum number of words, that JSON deliverables parse, and that the completion record has the required fields and a valid status. It knows nothing about the correct answers.

Recovery is controller-side and bounded. After the first session ends and its deliverables are collected, the controller runs the same self-check locally on the collected output. If it fails, or if the session ended by timeout, budget exhaustion or error, and the emergency stop is not set, the controller writes a recovery prompt into the workspace—stating that the previous session ended, quoting the self-check's problems, instructing the worker to read its checkpoint and the current output, finish the remaining deliverables, fix the reported problems, run the self-check and rewrite the completion record—and resumes the same session identifier with the same flags, the remaining budget and the remaining wall time. At most one recovery is granted per run. The recovery's cost is computed as the increment over the first session when the resumed client reports a cumulative figure, is added to the run's cost and recorded in the ledger under its own label, and its wall time is added to the run's total. The deliverables are collected again, and the final self-check status is recorded with the first; the two statuses are the data from which the Autonomous Recovery Index of Section 2.4 is estimated. All of a recovery's cost and time are attributed to the parent run, as the mandate requires for any sub-process within a treatment.

5.7. Logging, hashing, ledger, dashboard and emergency shutdown

Every session's raw event stream is written unmodified to an append-only file in the run's directory, with a separate file for the recovery stream; the normalised events are stored in a SQLite database on the controller together with the run table, the artefact table of paths, hashes and sizes, the receipts, the judgments, the metrics and the ledger. The ledger records every charge—worker, recovery, each judge—against the run that caused it, and its total is the quantity compared with the study envelope before each dispatch. The deterministic receipt carries the hash of the oracle and evaluator that

produced it; the workspace manifest carries the hashes of what the worker saw; the collection record carries the hashes of what it produced. The normalised run and judgment tables are exported to Parquet for the analysis, and the raw files are never rewritten.

A private dashboard is rendered from the database as a static page that refreshes itself: progress by arm and status, the ledger against the envelope, validated outcomes by arm, node utilisation, a list of errors and a table of runs with their status, node, result subtype, wall time, cost, tool calls, deterministic completion, violations, quality, acceptance and recovery flags. It is served on the controller's loopback interface only. Emergency shutdown is a single command that kills every bare-mode client and every portal process on every executor node and sets the stop event that prevents the scheduler from dispatching further runs or granting further recoveries; a stale-run detector lists runs that have been in a running state for longer than a threshold so that they can be killed and reset individually.

5.8. Scheduling and concurrency

The scheduler takes the assigned runs in the order of the seeded permutation, checks the ledger against the envelope, and dispatches each to the least-loaded node that is eligible for the arm and has a free slot, up to a global limit of concurrent sessions. The pilot's concurrency, three sessions per node and twelve in total, was chosen from the experience of an earlier study in the same series, which ran two sessions per node without incident on shorter runs, and from the longer expected duration of office and portal work; it is a judgement, not a measurement, and the infrastructure-stability statistics of the pilot are the test of it. Each dispatched run executes in its own thread on the controller: prepare, launch, collect, evaluate and judge, then release the node slot. A run that fails for an infrastructure reason before any inference occurs is marked as such with the exception recorded, its workspace is reset, and it is re-queued under the same identifier according to the exclusion rules of Section 4.11.

References

Abadie, Alberto, Susan Athey, Guido W Imbens, and Jeffrey M Wooldridge, "When Should You Adjust Standard Errors for Clustering?," *The Quarterly Journal of Economics*, 2023, *138* (1), 1–35. verified: crossref 2026-11-02. Crossref issued 2022-10-06 (online first); print volume 138(1), 2023.

Acemoglu, Daron, "The simple macroeconomics of AI," *Economic Policy*, 2025, *40* (121), 13–58. verified: crossref 2026-11-02.

_ **and David Autor**, "Skills, Tasks and Technologies: Implications for Employment and Earnings," in "Handbook of Labor Economics," Elsevier, 2011. verified: crossref 2026-11-02.

_ **and Pascual Restrepo**, "The Race between Man and Machine: Implications of Technology for Growth, Factor Shares, and Employment," *American Economic Review*, 2018, *108* (6), 1488–1542. verified: crossref 2026-11-02.

_ **and** _ , "Automation and New Tasks: How Technology Displaces and Reinstates Labor," *Journal of Economic Perspectives*, 2019, *33* (2), 3–30. verified: crossref 2026-11-02.

_ **and** _ , "Robots and Jobs: Evidence from US Labor Markets," *Journal of Political Economy*, 2020, *128* (6), 2188–2244. verified: crossref 2026-11-02.

Agarwal, Nikhil, Alex Moehring, Pranav Rajpurkar, and Tobias Salz, "Combining Human Expertise with Artificial Intelligence: Experimental Evidence from Radiology," NBER Working Paper 31422, National Bureau of Economic Research 2023. verified: crossref 2026-11-02.

Agrawal, Ajay, Joshua S. Gans, and Avi Goldfarb, "Artificial Intelligence: The Ambiguous Labor Market Impact of Automating Prediction," *Journal of Economic Perspectives*, 2019, *33* (2), 31–50. verified: crossref 2026-11-02.

Anthropic, "Introducing computer use, a new Claude 3.5 Sonnet, and Claude 3.5 Haiku," Anthropic announcement, 22 October 2024 October 2024. verified: web (official Anthropic page fetched 2026-11-02; H1 and datePublished 2024-10-22 match).

_ , "System Card: Claude Opus 4 & Claude Sonnet 4," Technical Report, Anthropic May 2025. verified: web (PDF fetched 2026-11-02; title page reads "System Card: Claude Opus 4 & Claude Sonnet 4, May 2025"; changelog to 2 Sept. 2025).

Appel, Ruth, Peter McCrory, Alex Tamkin, Miles McCain, Tyler Neylon, and Michael Stern, "Anthropic Economic Index report: Uneven geographic and enterprise AI adoption," arXiv preprint arXiv:2511.15080 2025. verified: arxiv 2511.15080.

Arntz, Melanie, Terry Gregory, and Ulrich Zierahn, "Revisiting the risk of automation," *Economics Letters*, 2017, *159*, 157–160. verified: crossref 2026-11-02.

Athey, Susan and Guido Imbens, "The Econometrics of Randomized Experiments," arXiv preprint arXiv:1607.00698 2016. verified: arxiv 1607.00698. Published as chapter 3 of the Handbook of Economic Field Experiments, vol. 1 (2017), doi:10.1016/bs.hefe.2016.10.003.

Athey, Susan C., Kevin A. Bryan, and Joshua S. Gans, "The Allocation of Decision Authority to Human and Artificial Intelligence," *AEA Papers and Proceedings*, 2020, *110*, 80–84. verified: crossref 2026-11-02.

Autor, D. H., F. Levy, and R. J. Murnane, "The Skill Content of Recent Technological Change: An Empirical Exploration," *The Quarterly Journal of Economics*, 2003, *118* (4), 1279–1333. verified: crossref 2026-11-02.

Autor, David H., "Why Are There Still So Many Jobs? The History and Future of Workplace Automation," *Journal of Economic Perspectives*, 2015, *29* (3), 3–30. verified: crossref 2026-11-02.

Backlund, Axel and Lukas Petersson, "Vending-Bench: A Benchmark for Long-Term Coherence of Autonomous Agents," arXiv preprint arXiv:2502.15840 2025. verified: arxiv 2502.15840.

Bates, Douglas, Martin Mächler, Ben Bolker, and Steve Walker, "Fitting Linear Mixed-Effects Models Using lme4," *Journal of Statistical Software*, 2015, *67* (1). verified: crossref 2026-11-02.

Becker, Joel, Nate Rush, Elizabeth Barnes, and David Rein, "Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity," arXiv preprint arXiv:2507.09089 2025. verified: arxiv 2507.09089.

Bowman, Samuel R., Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamilė Lukošiūtė, Amanda Askell, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Christopher Olah, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Jackson Kernion, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Liane Lovitt, Nelson Elhage, Nicholas Schiefer, Nicholas Joseph, Noemí Mercado, Nova DasSarma, Robin Larson, Sam McCandlish, Sandipan Kundu, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Ben Mann, and Jared Kaplan, "Measuring Progress on Scalable Oversight for Large Language Models," arXiv preprint arXiv:2211.03540 2022. verified: arxiv 2211.03540.

Brynjolfsson, Erik and Tom Mitchell, "What can machine learning do? Workforce implications," *Science*, 2017, *358* (6370), 1530–1534. verified: crossref 2026-11-02.

_ **, Bharat Chandar, and Ruyu Chen**, "Canaries in the Coal Mine? Six Facts about the Recent Employment Effects of Artificial Intelligence," Working Paper, Stanford Digital Economy Lab 2025. verified: web (official Stanford Digital Economy Lab page fetched 2026-11-02; authors and title match; first version August 2025, revised version dated 2026 on the page).

_ **, Danielle Li, and Lindsey Raymond**, "Generative AI at Work," *The Quarterly Journal of Economics*, 2025, *140* (2), 889–942. verified: crossref 2026-11-02.

_ **, Tom Mitchell, and Daniel Rock**, "What Can Machines Learn and What Does It Mean for Occupations and the Economy?," *AEA Papers and Proceedings*, 2018, *108*, 43–47. verified: crossref 2026-11-02.

Cameron, A. Colin and Douglas L. Miller, "A Practitioner's Guide to Cluster-Robust Inference," *Journal of Human Resources*, 2015, *50* (2), 317–372. verified: crossref 2026-11-02.

Cameron, A. Colin, Jonah B. Gelbach, and Douglas L. Miller, "Bootstrap-Based Improvements for Inference with Clustered Errors," *Review of Economics and Statistics*, 2008, *90* (3), 414–427. verified: crossref 2026-11-02.

Catalini, Christian, Xiang Hui, and Jane Wu, "Some Simple Economics of AGI," arXiv preprint arXiv:2602.20946 2026. verified: arxiv 2602.20946.

Cazzaniga, Mauro, Florence Jaumotte, Longji Li, Giovanni Melina, Augustus Panton, Carlo Pizzinelli, Emma Rockall, and Marina Mendes Tavares, "Gen-AI: Artificial Intelligence and the Future of Work," IMF Staff Discussion Note SDN/2024/001, International Monetary Fund 2024. verified: crossref 2026-11-02.

Cemri, Mert, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica, "Why Do Multi-Agent LLM Systems Fail?," arXiv preprint arXiv:2503.13657 2025. verified: arxiv 2503.13657.

Chan, Jun Shern, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Lilian Weng, and Aleksander Mądry, "MLE-bench: Evaluating Machine Learning Agents on Machine Learning Engineering," arXiv preprint arXiv:2410.07095 2024. verified: arxiv 2410.07095.

Chen, Xinyun, Maxwell Lin, Nathanael Schärli, and Denny Zhou, "Teaching Large Language Models to Self-Debug," arXiv preprint arXiv:2304.05128 2023. verified: arxiv 2304.05128.

Choi, Jonathan H., Amy Monahan, and Daniel Schwarcz, "Lawyering in the Age of Artificial Intelligence," SSRN working paper 2023. verified: crossref 2026-11-02.

Cox, D. R., "Regression Models and Life-Tables," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 1972, *34* (2), 187–202. verified: crossref 2026-11-02.

Cui, Zheyuan, Mert Demirer, Sonia Jaffe, Leon Musolff, Sida Peng, and Tobias Salz, "The Effects of Generative AI on High Skilled Work: Evidence from Three Field Experiments with Software Developers," SSRN working paper 2024. verified: crossref 2026-11-02.

Dell'Acqua, Fabrizio, Charles Ayoubi, Hila Lifshitz-Assaf, Raffaella Sadun, Ethan R. Mollick, Lilach Mollick, Yi Han, Jeff Goldman, Hari Nair, Stew Taub, and Karim R. Lakhani, "The Cybernetic Teammate: A Field Experiment on Generative AI Reshaping Teamwork and Expertise," Working Paper 25-043, Harvard Business School 2025. verified: crossref 2026-11-02. SSRN posted content.

_ , Edward McFowland III, Ethan R. Mollick, Hila Lifshitz-Assaf, Katherine Kellogg, Saran Rajendran, Lisa Krayer, François Candelon, and Karim R. Lakhani, "Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of Artificial Intelligence on Knowledge Worker Productivity and Quality," Working Paper 24-013, Harvard Business School 2023. verified: crossref 2026-11-02. SSRN posted content.

Dietvorst, Berkeley J., Joseph P. Simmons, and Cade Massey, "Algorithm aversion: People erroneously avoid algorithms after seeing them err.," *Journal of Experimental Psychology: General*, 2015, *144* (1), 114–126. verified: crossref 2026-11-02.

Drouin, Alexandre, 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?," arXiv preprint arXiv:2403.07718 2024. verified: arxiv 2403.07718.

Dziri, Nouha, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi, "Faith and Fate: Limits of Transformers on Compositionality," arXiv preprint arXiv:2305.18654 2023. verified: arxiv 2305.18654.

Eloundou, Tyna, Sam Manning, Pamela Mishkin, and Daniel Rock, "GPTs are GPTs: Labor market impact potential of LLMs," *Science*, 2024, *384* (6702), 1306–1308. verified: crossref 2026-11-02.

Felten, Ed, Manav Raj, and Robert Seamans, "How will Language Modelers like ChatGPT Affect Occupations and Industries?," arXiv preprint arXiv:2303.01157 2023. verified: arxiv 2303.01157.

Felten, Edward, Manav Raj, and Robert Seamans, "Occupational, industry, and geographic exposure to artificial intelligence: A novel dataset and its potential uses," *Strategic Management Journal*, 2021, *42* (12), 2195–2217. verified: crossref 2026-11-02.

Fine, Jason P. and Robert J. Gray, "A Proportional Hazards Model for the Subdistribution of a Competing Risk," *Journal of the American Statistical Association*, 1999, *94* (446), 496–509. verified: crossref 2026-11-02.

Frey, Carl Benedikt and Michael A. Osborne, "The future of employment: How susceptible are jobs to computerisation?," *Technological Forecasting and Social Change*, 2017, *114*, 254–280. verified: crossref 2026-11-02.

Gmyrek, Pawel, Janine Berg, and David Bescond, "Generative AI and jobs: A global analysis of potential effects on job quantity and quality," ILO Working Paper 96, International Labour Organization, Geneva 2023. verified: crossref 2026-11-02.

_ , _ , Karol Kamiński, Filip Konopczyński, Agnieszka Ładna, Balint Nafradi, Konrad Rosłaniec, and Marek Troszyński, "Generative AI and jobs: A refined global index of occupational exposure," ILO Working Paper 140, International Labour Organization, Geneva 2025. verified: crossref 2026-11-02.

Gu, Jiawei, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo, "A Survey on LLM-as-a-Judge," arXiv preprint arXiv:2411.15594 2024. verified: arxiv 2411.15594.

Guha, Neel, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad

Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li, "LegalBench: A Collaboratively Built Benchmark for Measuring Legal Reasoning in Large Language Models," arXiv preprint arXiv:2308.11462 2023. verified: arxiv 2308.11462.

Gwet, Kilem Li, "Computing inter-rater reliability and its variance in the presence of high agreement," *British Journal of Mathematical and Statistical Psychology*, 2008, *61* (1), 29–48. verified: crossref 2026-11-02.

Hadfield, Gillian K. and Andrew Koh, "An Economy of AI Agents," arXiv preprint arXiv:2509.01063 2025. verified: arxiv 2509.01063.

Hampole, Menaka, Dimitris Papanikolaou, Lawrence Schmidt, and Bryan Seegmiller, "Artificial Intelligence and the Labor Market," NBER Working Paper 33509, National Bureau of Economic Research 2025. verified: crossref 2026-11-02.

Handa, Kunal, Alex Tamkin, Miles McCain, Saffron Huang, Esin Durmus, Sarah Heck, Jared Mueller, Jerry Hong, Stuart Ritchie, Tim Belonax, Kevin K. Troy, Dario Amodei, Jared Kaplan, Jack Clark, and Deep Ganguli, "Which Economic Tasks are Performed with AI? Evidence from Millions of Claude Conversations," arXiv preprint arXiv:2503.04761 2025. verified: arxiv 2503.04761.

Hayes, Andrew F. and Klaus Krippendorff, "Answering the Call for a Standard Reliability Measure for Coding Data," *Communication Methods and Measures*, 2007, *1* (1), 77–89. verified: crossref 2026-11-02.

Holmstrom, Bengt, "Moral Hazard and Observability," *The Bell Journal of Economics*, 1979, *10* (1), 74. verified: crossref 2026-11-02.

_ **and Paul Milgrom**, "Multitask Principal–Agent Analyses: Incentive Contracts, Asset Ownership, and Job Design," *The Journal of Law, Economics, and Organization*, 1991, *7* (special_issue), 24–52. verified: crossref 2026-11-02.

Horton, John, Apostolos Filippas, and Benjamin Manning, "Large Language Models as Simulated Economic Agents: What Can We Learn from Homo Silicus?," NBER Working Paper 31122, National Bureau of Economic Research 2023. verified: crossref 2026-11-02.

Huang, Jie, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou, "Large Language Models Cannot Self-Correct Reasoning Yet," arXiv preprint arXiv:2310.01798 2023. verified: arxiv 2310.01798.

Humlum, Anders and Emilie Vestergaard, "Still Waters, Rapid Currents: Early Labor Market Transformation under Generative AI," NBER Working Paper 33777, National Bureau of Economic Research 2025. verified: crossref 2026-11-02. Previously circulated as "Large Language Models, Small Labor Market Effects" (May 2025).

Ide, Enrique and Eduard Talamàs, "Artificial Intelligence in the Knowledge Economy," *Journal of Political Economy*, 2025, *133* (12), 3762–3800. verified: crossref 2026-11-02.

Immorlica, Nicole, Brendan Lucier, and Aleksandrs Slivkins, "Generative AI as Economic Agents," arXiv preprint arXiv:2406.00477 2024. verified: arxiv 2406.00477.

Jimenez, Carlos E., John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan, "SWE-bench: Can Language Models Resolve Real-World GitHub Issues?," arXiv preprint arXiv:2310.06770 2023. verified: arxiv 2310.06770.

Kaplan, E. L. and Paul Meier, "Nonparametric Estimation from Incomplete Observations," *Journal of the American Statistical Association*, 1958, *53* (282), 457–481. verified: crossref 2026-11-02.

Kapoor, Sayash, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan, "AI Agents That Matter," arXiv preprint arXiv:2407.01502 2024. verified: arxiv 2407.01502.

Kolt, Noam, "Governing AI Agents," arXiv preprint arXiv:2501.07913 2025. verified: arxiv 2501.07913.

Korinek, Anton, "Generative AI for Economic Research: Use Cases and Implications for Economists," *Journal of Economic Literature*, 2023, *61* (4), 1281–1317. verified: crossref 2026-11-02.

Krippendorff, Klaus, "Reliability in Content Analysis: Some Common Misconceptions and Recommendations," *Human Communication Research*, 2004, *30* (3), 411–433. verified: crossref 2026-11-02.

Kwa, Thomas, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, Sami Jawhar, Megan Kinniment, Nate Rush, Sydney Von Arx, Ryan Bloom, Thomas Broadley, Haoxing Du, Brian Goodrich, Nikola Jurkovic, Luke Harold Miles, Seraphina Nix, Tao Lin, Chris Painter, Neev Parikh, David Rein, Lucas Jun Koba Sato, Hjalmar Wijk, Daniel M. Ziegler, Elizabeth Barnes, and Lawrence Chan, "Measuring AI Ability to Complete Long Software Tasks," arXiv preprint arXiv:2503.14499 2025. verified: arxiv 2503.14499.

List, John A., Azeem M. Shaikh, and Yang Xu, "Multiple hypothesis testing in experimental economics," *Experimental Economics*, 2019, *22* (4), 773–793. verified: crossref 2026-11-02.

Liu, Jiacheng, Xiaohan Zhao, Xinyi Shang, and Zhiqiang Shen, "Dive into Claude Code: The Design Space of Today's and Future AI Agent Systems," arXiv preprint arXiv:2604.14228 2026. verified: arxiv 2604.14228.

Liu, Xiao, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang, "AgentBench: Evaluating LLMs as Agents," arXiv preprint arXiv:2308.03688 2023. verified: arxiv 2308.03688.

Ma, Zeyao, Bohan Zhang, Jing Zhang, Jifan Yu, Xiaokang Zhang, Xiaohan Zhang, Sijia Luo, Xi Wang, and Jie Tang, "SpreadsheetBench: Towards Challenging Real World Spreadsheet Manipulation," arXiv preprint arXiv:2406.14991 2024. verified: arxiv 2406.14991.

Madaan, Aman, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark, "Self-Refine: Iterative Refinement with Self-Feedback," arXiv preprint arXiv:2303.17651 2023. verified: arxiv 2303.17651.

Merrill, Mike A., Alexander G. Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E. Kelly Buchanan, Junhong Shen, Guanghao Ye, Haowei Lin, Jason Poulos, Maoyu Wang, Marianna Nezhurina, Jenia Jitsev, Di Lu, Orfeas Menis Mastromichalakis, Zhiwei Xu, Zizhao Chen, Yue Liu, Robert Zhang, Leon Liangyu Chen, Anurag Kashyap, Jan-Lucas Uslu, Jeffrey Li, Jianbo Wu, Minghao Yan, Song Bian, Vedang Sharma, Ke Sun, Steven Dillmann, Akshay Anand, Andrew Lanpouthakoun, Bardia Koopah, Changran Hu, Etash Guha, Gabriel H. S. Dreiman, Jiacheng Zhu, Karl Krauth, Li Zhong, Niklas Muennighoff, Robert Amanfu, Shangyin Tan, Shreyas Pimpalgaonkar, Tushar Aggarwal, Xiangning Lin, Xin Lan, Xuandong Zhao, Yiqing Liang, Yuanli Wang, Zilong Wang, Changzhi Zhou, David Heineman, Hange Liu, Harsh Trivedi, John Yang, Junhong Lin, Manish Shetty, Michael Yang, Nabil Omi, Negin Raoof, Shanda Li, Terry Yue Zhuo, Wuwei Lin, Yiwei Dai, Yuxin Wang, Wenhao Chai, Shang Zhou, Dariush Wahdany, Ziyu She, Jiaming Hu, Zhikang Dong, Yuxuan Zhu, Sasha Cui, Ahson Saiyed, Arinbjörn Kolbeinsson, Jesse Hu, Christopher Michael Rytting, Ryan Marten, Yixin Wang, Alex Dimakis, Andy Konwinski, and Ludwig Schmidt, "Terminal-Bench: Benchmarking

Agents on Hard, Realistic Tasks in Command Line Interfaces," arXiv preprint arXiv:2601.11868 2026. verified: arxiv 2601.11868.

Mialon, Grégoire, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom, "GAIA: a benchmark for General AI Assistants," arXiv preprint arXiv:2311.12983 2023. verified: arxiv 2311.12983.

Miller, Evan, "Adding Error Bars to Evals: A Statistical Approach to Language Model Evaluations," arXiv preprint arXiv:2411.00640 2024. verified: arxiv 2411.00640.

Miserendino, Samuel, Michele Wang, Tejal Patwardhan, and Johannes Heidecke, "SWE-Lancer: Can Frontier LLMs Earn $1 Million from Real-World Freelance Software Engineering?," arXiv preprint arXiv:2502.12115 2025. verified: arxiv 2502.12115.

Nosek, Brian A., Charles R. Ebersole, Alexander C. DeHaven, and David T. Mellor, "The preregistration revolution," *Proceedings of the National Academy of Sciences*, 2018, *115* (11), 2600–2606. verified: crossref 2026-11-02.

Noy, Shakked and Whitney Zhang, "Experimental evidence on the productivity effects of generative artificial intelligence," *Science*, 2023, *381* (6654), 187–192. verified: crossref 2026-11-02.

Olken, Benjamin A., "Promises and Perils of Pre-Analysis Plans," *Journal of Economic Perspectives*, 2015, *29* (3), 61–80. verified: crossref 2026-11-02.

Otis, Nicholas, Rowan Clarke, Solène Delecourt, David Holtz, and Rembrand Koning, "The Uneven Impact of Generative AI on Entrepreneurial Performance," SSRN working paper 2024. verified: crossref 2026-11-02.

Panickssery, Arjun, Samuel R. Bowman, and Shi Feng, "LLM Evaluators Recognize and Favor Their Own Generations," arXiv preprint arXiv:2404.13076 2024. verified: arxiv 2404.13076.

Patwardhan, Tejal, 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," arXiv preprint arXiv:2510.04374 2025. verified: arxiv 2510.04374.

Peng, Sida, Eirini Kalliamvakou, Peter Cihon, and Mert Demirer, "The Impact of AI on Developer Productivity: Evidence from GitHub Copilot," arXiv preprint arXiv:2302.06590 2023. verified: arxiv 2302.06590.

Pizzinelli, Carlo, Augustus Panton, Marina Mendes Tavares, Mauro Cazzaniga, and Longji Li, "Labor Market Exposure to AI: Cross-country Differences and Distributional Implications," IMF Working Paper WP/23/216, International Monetary Fund 2023. verified: crossref 2026-11-02.

Polo, Felipe Maia, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin, "tinyBenchmarks: evaluating LLMs with fewer examples," arXiv preprint arXiv:2402.14992 2024. verified: arxiv 2402.14992.

Rein, David, Joel Becker, Amy Deng, Seraphina Nix, Chris Canal, Daniel O'Connel, Pip Arnott, Ryan Bloom, Thomas Broadley, Katharyn Garcia, Brian Goodrich, Max Hasin, Sami Jawhar, Megan Kinniment, Thomas Kwa, Aron Lajko, Nate Rush, Lucas Jun Koba Sato, Sydney Von Arx, Ben West, Lawrence Chan, and Elizabeth Barnes, "HCAST: Human-Calibrated Autonomy Software Tasks," arXiv preprint arXiv:2503.17354 2025. verified: arxiv 2503.17354.

Restrepo, Pascual, "We Won't be Missed: Work and Growth in the AGI World," NBER Working Paper 34423, National Bureau of Economic Research 2025. verified: crossref 2026-11-02.

Shinn, Noah, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao, "Reflexion: Language Agents with Verbal Reinforcement Learning," arXiv preprint arXiv:2303.11366 2023. verified: arxiv 2303.11366.

Simmons, Joseph P., Leif D. Nelson, and Uri Simonsohn, "False-Positive Psychology," *Psychological Science*, 2011, *22* (11), 1359–1366. verified: crossref 2026-11-02.

Sinha, Akshit, Arvindh Arun, Shashwat Goel, Steffen Staab, and Jonas Geiping, "The Illusion of Diminishing Returns: Measuring Long Horizon Execution in LLMs," arXiv preprint arXiv:2509.09677 2025. verified: arxiv 2509.09677.

Sreenivas, Nanda Kishore and Kate Larson, "Contracting for LLM Delegation: Moral Hazard in Technology and Effort Choice," arXiv preprint arXiv:2608.18232 2026. verified: arxiv 2608.18232.

Stroebl, Benedikt, Sayash Kapoor, and Arvind Narayanan, "The Limits of Inference Scaling Through Resampling," arXiv preprint arXiv:2411.17501 2024. verified: arxiv 2411.17501.

Svanberg, Maja, Wensu Li, Martin Fleming, Brian Goehring, and Neil Thompson, "Beyond AI Exposure: Which Tasks are Cost-Effective to Automate with Computer Vision?," SSRN working paper 2024. verified: crossref 2026-11-02.

Tomlinson, Kiran, Sonia Jaffe, Will Wang, Scott Counts, and Siddharth Suri, "Working with AI: Measuring the Applicability of Generative AI to Occupations," arXiv preprint arXiv:2507.07935 2025. verified: arxiv 2507.07935.

Trivedi, Harsh, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian, "AppWorld: A Controllable World of Apps and People for Benchmarking Interactive Coding Agents," in "Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)" 2024. verified: crossref 2026-11-02.

Vaccaro, Michelle, Abdullah Almaatouq, and Thomas Malone, "When combinations of humans and AI are useful: A systematic review and meta-analysis," *Nature Human Behaviour*, 2024, *8* (12), 2293–2303. verified: crossref 2026-11-02.

Vania, Clara, Phu Mon Htut, William Huang, Dhara Mungra, Richard Yuanzhe Pang, Jason Phang, Haokun Liu, Kyunghyun Cho, and Samuel R. Bowman, "Comparing Test Sets with Item Response Theory," in "Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)" 2021. verified: crossref 2026-11-02.

Wang, Peiyi, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui, "Large Language Models are not Fair Evaluators," in "Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)" 2024. verified: crossref 2026-11-02.

Wang, Xingyao, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Yanjun Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, Robert Brennan, Hao Peng, Heng Ji, and Graham Neubig, "OpenHands: An Open Platform for AI Software Developers as Generalist Agents," arXiv preprint arXiv:2407.16741 2024. verified: arxiv 2407.16741.

Wang, Zilong, Yuedong Cui, Li Zhong, Zimin Zhang, Da Yin, Bill Yuchen Lin, and Jingbo Shang, "OfficeBench: Benchmarking Language Agents across Multiple Applications for Office Automation," arXiv preprint arXiv:2407.19056 2024. verified: arxiv 2407.19056.

Wijk, Hjalmar, Tao Lin, Joel Becker, Sami Jawhar, Neev Parikh, Thomas Broadley, Lawrence Chan, Michael Chen, Josh Clymer, Jai Dhyani, Elena Ericheva, Katharyn Garcia, Brian Goodrich, Nikola Jurkovic, Holden Karnofsky, Megan Kinniment, Aron Lajko, Seraphina

Nix, Lucas Sato, William Saunders, Maksym Taran, Ben West, and Elizabeth Barnes, "RE-Bench: Evaluating frontier AI R&D capabilities of language model agents against human experts," arXiv preprint arXiv:2411.15114 2024. verified: arxiv 2411.15114.

Xie, Tianbao, 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," arXiv preprint arXiv:2404.07972 2024. verified: arxiv 2404.07972.

Xu, Frank F., 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," arXiv preprint arXiv:2412.14161 2024. verified: arxiv 2412.14161.

Yang, John, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press, "SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering," arXiv preprint arXiv:2405.15793 2024. verified: arxiv 2405.15793.

Yao, Shunyu, Noah Shinn, Pedram Razavi, and Karthik Narasimhan, "τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains," arXiv preprint arXiv:2406.12045 2024. verified: arxiv 2406.12045.

Yuan, Mengqi, Zilong Zhou, Xinzhuang Xiong, Weiming Wu, Jiayang Sun, Jiamin Song, Kaiqian Cui, Bowen Wang, Haoyuan Wu, Yitong Li, Dunjie Lu, Haikong Lu, Qi Zhen, Xinyuan Wang, Jiaqi Deng, Yuhao Yang, Cheng Chen, Boyuan Zheng, Alex Su, Xiao Yu, Hao Zou, Saaket Agashe, Xing Han Lu, Manpreet Kaur, Zhengyang Qi, Vincent Sunn Chen, Frederic Sala, Dayiheng Liu, Junyang Lin, Zhou Yu, Yu Su, Siva Reddy, Xin Eric Wang, Peng Qi, Tianbao Xie, and Tao Yu, "OSWorld 2.0: Benchmarking Computer Use Agents on Long-Horizon Real-World Tasks," arXiv preprint arXiv:2606.29537 2026. verified: arxiv 2606.29537.

Zheng, Lianmin, 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," arXiv preprint arXiv:2306.05685 2023. verified: arxiv 2306.05685.

Zhou, Hongli, Hui Huang, Ziqing Zhao, Lvyuan Han, Huicheng Wang, Kehai Chen, Muyun Yang, Wei Bao, Jian Dong, Bing Xu, Conghui Zhu, Hailong Cao, and Tiejun Zhao, "Lost in

Benchmarks? Rethinking Large Language Model Benchmarking with Item Response Theory," arXiv preprint arXiv:2505.15055 2025. verified: arxiv 2505.15055.

Zhou, Shuyan, 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," arXiv preprint arXiv:2307.13854 2023. verified: arxiv 2307.13854.

Zhu, Yuxuan, Tengjun Jin, Yada Pruksachatkun, Andy Zhang, Shu Liu, Sasha Cui, Sayash Kapoor, Shayne Longpre, Kevin Meng, Rebecca Weiss, Fazl Barez, Rahul Gupta, Jwala Dhamala, Jacob Merizian, Mario Giulianelli, Harry Coppock, Cozmin Ududec, Jasjeet Sekhon, Jacob Steinhardt, Antony Kellermann, Sarah Schwettmann, Matei Zaharia, Ion Stoica, Percy Liang, and Daniel Kang, "Establishing Best Practices for Building Rigorous Agentic Benchmarks," arXiv preprint arXiv:2507.02825 2025. verified: arxiv 2507.02825.

A. Mathematical appendix

This appendix proves the propositions stated in Section 2 and records two results used by the inference plan of Section 4.9: the variance of the scenario-weighted estimator under cluster sampling and a sketch of the validity of the scenario-cluster bootstrap. Notation is that of Section 2; the occupation index o is suppressed when it plays no role.

A.1. Proof of Proposition 1 (properties of the frontier)

Write $\pi_j(B,H,q) = \Pr\{S_j(q)=1, C_j \le B, H_j \le H\}$ so that $\mathrm{ALF}(B,H,q) = \sum_j w_j \pi_j(B,H,q)$.

(i) Each π_j is a probability, hence in $[0,1]$; the weights are non-negative and sum to one, so the weighted sum lies in $[0,1]$.

(ii) For $B \le B'$ the event $\{C_j \le B\}$ is contained in $\{C_j \le B'\}$, hence $\pi_j(B,H,q) \le \pi_j(B',H,q)$; the same holds in H. For $q \le q'$, $\{Q_j \ge q'\} \subseteq \{Q_j \ge q\}$, hence $S_j(q') \le S_j(q)$ pointwise and π_j is non-increasing in q. Right-continuity in B follows from the continuity from above of probability measures: for $B_n \downarrow B$, $\bigcap_n \{C_j \le B_n\} = \{C_j \le B\}$, so $\pi_j(B_n,H,q) \to \pi_j(B,H,q)$; likewise in H. Finite weighted sums preserve these properties.

(iii) $\{S_j(q)=1, C_j \le B, H_j \le H\} \subseteq \{S_j(q)=1\}$ gives $\pi_j \le p_j(q)$ and hence $\mathrm{ALF} \le \rho(q)$. For $B_n \uparrow \infty$ and $H_n \uparrow \infty$, $\bigcup_n \{C_j \le B_n, H_j \le H_n\} = \{C_j < \infty, H_j < \infty\}$, which has probability one because C_j and H_j are finite random variables; continuity from below gives $\pi_j \to p_j(q)$ and the limit of the weighted sum follows.

(iv) For events A_1, A_2, A_3 in a probability space, $\Pr(A_1 \cap A_2 \cap A_3) \le \min_k \Pr(A_k)$ and, by Bonferroni's inequality applied twice, $\Pr(A_1 \cap A_2 \cap A_3) \ge \Pr(A_1) + \Pr(A_2) + \Pr(A_3) - 2$. Apply with $A_1 = \{S_j(q)=1\}$, $A_2 = \{C_j \le B\}$, $A_3 = \{H_j \le H\}$, take the maximum with zero on the lower side, and sum with the weights.

(v) If every run is stopped when its cost reaches $\bar{B}$ and evaluated as delivered, then $C_j \le \bar{B}$ almost surely, so $\{C_j \le B\} = \{C_j \le \bar{B}\}$ up to a null set for every $B \ge \bar{B}$, and $\pi_j(B,H,q) = \pi_j(\bar{B},H,q)$.

(vi) Since $\sum_j w_j = \sum_j w'_j = 1$, for any constant c,

$$\mathrm{ALF}^{w} - \mathrm{ALF}^{w'} = \sum_j (w_j - w'_j)\pi_j = \sum_j (w_j - w'_j)(\pi_j - c).$$

Choose $c = \tfrac{1}{2}(\max_j \pi_j + \min_j \pi_j)$, so that $|\pi_j - c| \le \tfrac{1}{2}(\max_j \pi_j - \min_j \pi_j)$ for every j. Then

$$\left|\mathrm{ALF}^{w} - \mathrm{ALF}^{w'}\right| \le \sum_j |w_j - w'_j| \cdot \tfrac{1}{2}(\max_j \pi_j - \min_j \pi_j) = \mathrm{TV}(w,w')\,(\max_j \pi_j - \min_j \pi_j). \qquad \square$$

A.2. Proof of Proposition 2 (decomposition)

By Definition 1, $\{S_j(q)=1\}=\{D_j=1\}\cap\{Q_j\geq q\}\cap\{E_j=0\}$. The chain rule of conditional probability gives, whenever each conditioning event has positive probability,

$$\begin{aligned}\Pr\{S_j(q)=1, C_j\leq B, H_j\leq H\} &= \Pr\{D_j=1\}\,\Pr\{Q_j\geq q\mid D_j=1\}\\ &\quad\times\Pr\{E_j=0\mid D_j=1, Q_j\geq q\}\,\Pr\{C_j\leq B, H_j\leq H\mid S_j(q)=1\}.\end{aligned}$$

If a conditioning event has probability zero, the left-hand side is zero and the convention that the corresponding factor is zero preserves the identity. Weighting by w_j and summing over j gives the frontier. □

A.3. Proof of Proposition 3 (identification and variance of the premium)

Under Assumption 1, $S_{j1}(a),\ldots,S_{jR_j}(a)$ are i.i.d. Bernoulli with mean $p_j(a)$, so $\mathbb{E}[\bar S_j(a)]=p_j(a)$ and $\mathrm{Var}[\bar S_j(a)]=p_j(a)\{1-p_j(a)\}/R_j$. Runs under different arms are separate sessions in separate workspaces and are independent by the no-interference clause, so $\mathrm{Cov}[\bar S_j(a),\bar S_j(a')]=0$; runs on different tasks are likewise independent. Linearity of expectation gives $\mathbb{E}[\hat\tau]=\sum_j w_j\{p_j(a)-p_j(a')\}=\tau$, and independence across tasks and arms gives (7). Assumption 2 is what makes τ interpretable as the effect of the tool bundle rather than of information, since it ensures that the potential outcomes under the two arms differ only through the tool configuration.

For (8), let the tasks be an i.i.d. sample of size J from a task population and let $w_j=1/J$. Write $\hat\Delta_j=\bar S_j(a)-\bar S_j(a')$, so that $\hat\tau=J^{-1}\sum_j\hat\Delta_j$ with the $\hat\Delta_j$ i.i.d. across j. By the law of total variance, conditioning on the task's parameters,

$$\mathrm{Var}(\hat\Delta_j)=\mathrm{Var}_j\big(\mathbb{E}[\hat\Delta_j\mid j]\big)+\mathbb{E}_j\big(\mathrm{Var}[\hat\Delta_j\mid j]\big)=\mathrm{Var}_j(\Delta_j)+\mathbb{E}_j\Big(\frac{\sigma_j^2(a)+\sigma_j^2(a')}{R_j}\Big),$$

and $\mathrm{Var}(\hat\tau)=J^{-1}\mathrm{Var}(\hat\Delta_j)$. □

The decomposition shows that the second term is divided by R_j and the first is not: repeats shrink the within-task component only, and the between-task component $\mathrm{Var}_j(\Delta_j)$ shrinks only with J. This is the formal reason for the primary inference to resample tasks rather than runs.

A.4. Proof of Proposition 4 (compounding reliability)

(i) With $\pi_k=\pi$, $p(d)=\pi^d$ and $\log p(d)=d\log\pi$, linear in d with slope $\log\pi<0$ for $\pi<1$. Setting $\pi^d=\frac12$ gives $d_{1/2}=\log 2/\log(1/\pi)$. The approximation $d_{1/2}\approx 0.69/(1-\pi)$ for π near one follows from $\log(1/\pi)=-\log\{1-(1-\pi)\}=(1-\pi)+O((1-\pi)^2)$.

(ii) By the inequality of arithmetic and geometric means, $\left(\prod_{k=1}^{d}\pi_k\right)^{1/d} \le d^{-1}\sum_{k=1}^{d}\pi_k = \bar{\pi}$, with equality if and only if all π_k are equal; raising to the power d gives $p(d) \le \bar{\pi}^d$.

(iii) $-\log p(d) = \sum_{k=1}^{d}(-\log\pi_k)$. Its first difference in d is $-\log\pi_{d+1}$, which is non-decreasing in d when π_k is non-increasing in k; a function on the integers with non-decreasing first differences is convex. □

A.5. Proof of Proposition 5 (recovery)

(i) A step is completed if it succeeds at the first attempt (probability π) or fails, is detected and is corrected (probability $(1-\pi)\delta\gamma = (1-\pi)\rho$). The events are disjoint, so the effective step reliability is $\pi' = \pi + (1-\pi)\rho \ge \pi$, with strict inequality when $\pi < 1$ and $\rho > 0$. With homogeneous steps, $p_\rho(d) = (\pi')^d$ by (9), and

$$\frac{p_\rho(d)}{p(d)} = \left(\frac{\pi + (1-\pi)\rho}{\pi}\right)^d = \left(1 + \rho\,\frac{1-\pi}{\pi}\right)^d,$$

which is geometric in d with ratio greater than one.

(ii) Let F be the event that the first session does not produce an accepted deliverable; under the stated distributional assumption $\Pr(F) = 1 - p(A3;d)$. On F, the self-check flags the run with probability δ and, if flagged, the recovery session produces an accepted deliverable with probability γ; on F^c the run is already accepted. By the law of total probability, $p(A4;d) = p(A3;d) + (1-p(A3;d))\,\delta\gamma$, and the difference is $(1-p(A3;d))\delta\gamma$, which is non-negative and increasing in $1-p(A3;d)$. □

The bound stated after Definition 4 follows by noting that $\gamma \le \Pr\{P_2 \mid F_1\} = \mathrm{ARI}$ when acceptance implies passing the self-check, and that $\delta \le 1$; hence $\tau(A4,A3) \le \Pr(F_1)\cdot\mathrm{ARI}$ under the same distributional assumption.

A.6. Proof of Proposition 6 (delegation thresholds)

The expected payoffs are: in-house, $U_H = v - c_H$; delegation with verification, $U_V = v - c_A - \kappa_V - (1-p)c_H$, since the deliverable is accepted with probability p and otherwise the failure is detected and the task is completed in-house at cost c_H; delegation without verification, $U_N = v - c_A - (1-p)\ell - eL$, since an unacceptable deliverable is used at loss ℓ and a critical error at loss L.

(i) $U_V \ge U_H \iff -c_A - \kappa_V - (1-p)c_H \ge -c_H \iff p\,c_H \ge c_A + \kappa_V \iff p \ge (c_A + \kappa_V)/c_H$.

(ii) $U_V \ge U_N \iff -\kappa_V - (1-p)c_H \ge -(1-p)\ell - eL \iff \kappa_V \le (1-p)(\ell - c_H) + eL$.

(iii) Delegation without verification is optimal when $U_N > U_V$ and $U_N \ge U_H$; the first is the negation of (ii) and the second is $c_A + (1-p)\ell + eL \le c_H$. As $L \to \infty$ with $e > 0$, the left-hand side

of the second condition diverges, so the condition fails; the right-hand side of (ii) also diverges, so verification is preferred to no verification, and the choice reduces to (i). □

A.7. Variance of the scenario-weighted estimator under cluster sampling

Let the occupation's tasks be sampled i.i.d. from a task population, let each sampled task j receive R_j i.i.d. runs, and let $g(Y)$ be any bounded run-level statistic (acceptance, acceptance within budget, cost, a product of indicators). The scenario-weighted estimator of $\theta = \mathbb{E}_j[\mu_j]$, with $\mu_j = \mathbb{E}[g(Y_{jr}) \mid j]$, is $\hat{\theta} = \sum_j w_j \bar{g}_j$, $\bar{g}_j = R_j^{-1} \sum_r g(Y_{jr})$. With general fixed weights w_j summing to one and $\sigma_j^2 = \mathrm{Var}[g(Y_{jr}) \mid j]$,

$$\mathbb{E}[\hat{\theta}] = \sum_j w_j \mathbb{E}_j[\mu_j] = \theta, \qquad \mathrm{Var}[\hat{\theta}] = \sum_j w_j^2 \left[\mathrm{Var}_j(\mu_j) + \mathbb{E}_j\left(\frac{\sigma_j^2}{R_j}\right) \right] + \sum_{j \neq j'} w_j w_{j'} \mathrm{Cov}(\bar{g}_j, \bar{g}_{j'}),$$

and the covariance term vanishes under i.i.d. task sampling. For equal weights this is (8) with g in place of the paired difference. For professional-minutes weights $w_j \propto m_j$ the effective number of tasks is $J_{\mathrm{eff}} = 1/\sum_j w_j^2 \leq J$, which is the price of unequal weighting. A consistent estimator of the variance is the scenario-level sample variance of $\bar{g}_j$ divided by J (equal weights), or its weighted analogue, which is what the cluster-robust standard errors of the sensitivity regressions estimate (Colin Cameron and Miller, 2015).

A.8. Validity of the scenario-cluster bootstrap: a sketch

The primary interval of Section 4.9 resamples scenarios with replacement, keeping each scenario's runs together, and recomputes the scenario-weighted statistic. Validity rests on three observations. First, under i.i.d. task sampling the scenario-level summaries $\hat{\Delta}_1, \ldots, \hat{\Delta}_J$ (for a contrast) or $\bar{g}_1, \ldots, \bar{g}_J$ (for a level) are i.i.d. bounded random variables, so the estimator is a sample mean of i.i.d. bounded terms and the ordinary nonparametric bootstrap of the mean is consistent for its sampling distribution as $J \to \infty$; keeping runs together within the resampled scenario preserves the within-scenario dependence that resampling runs would destroy (Cameron et al., 2008; Abadie et al., 2023). Second, for ratio estimands such as the cost per accepted deliverable or the Autonomy-Adjusted Productivity, $\hat{\theta} = \sum_j w_j \bar{u}_j / \sum_j w_j \bar{v}_j$, the delta method shows that $\hat{\theta}$ is asymptotically linear in the scenario-level pairs $(\bar{u}_j, \bar{v}_j)$, and the same bootstrap, applied to the pairs, is consistent provided the denominator's limit is bounded away from zero; this is why numerator and denominator are resampled jointly. Third, the paired sign-flip permutation test is exact in finite samples under the sharp null that the arm has no effect on any run of any scenario, because under that null the sign of each scenario's difference is exchangeable; it therefore complements the bootstrap at the pilot's $J = 10$, where asymptotic approximations are rough and percentile intervals may undercover. With few clusters,

the wild cluster bootstrap of Cameron et al. (2008) is the recommended refinement for the regression-based sensitivity analyses, and the preregistration's statement that intervals at $J = 10$ will be wide and reported as such is the honest consequence of these approximations.