UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

DÉPARTEMENT DES SCIENCES ADMINISTRATIVES

WORKING PAPER NO. 42

The Autonomous White-Collar Worker: A Randomised Tool-Access Experiment on Claude Code Across Ten Occupations

Journal version of UQO Working Paper No. 42

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

We report a preregistered randomised experiment on whether an autonomous computer-using agent can complete professional assignments to an acceptable standard. A controller launches independent Claude Code sessions (`claude-opus-5-5`) to perform ten validated scenarios in ten occupations inside a synthetic economy with hidden oracles, planted errors and prompt-injection traps. Each scenario is run under five levels of computer access, from file access without a shell to full computer use with browser, office tools and an enterprise portal, plus a self-repairing variant. Acceptance requires deterministic completion, no critical violation and a blind quality score of at least 70. Over 150 runs, acceptance is 0.00, 0.87, 0.90, 0.90 and 0.83. The shell premium is $+0.87$ [0.67, 1.00]; the preregistered primary contrast, full computer use minus shell, is $+0.03$ [0.00, 0.10], a precise null because every arm with a shell is at a ceiling. Accepted deliverables cost about one dollar of API charges; a cheaper model reaches the same acceptance at 0.40 dollars. In an exploratory experiment, bounded self-repair converts interrupted sessions into accepted deliverables (0.85 against 0.05). The supervision margin, not the budget, separates the Autonomous Labor Frontier from the acceptance ceiling. Human arms were not executed; no displacement estimate is implied.

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

1. Introduction

Large language models have become agents. A model that once answered questions in a chat window can now be given a shell, a file system, a browser and a set of office applications, told to complete an assignment, and left alone until it reports back. Claude Code, the command-line client studied here, is one such agent: a loop in which the model reads files, runs programs, inspects their output, edits documents and decides what to do next, with no human in the loop unless the configuration requires one (Liu et al., 2026; Anthropic, 2024). The economic question this raises is not whether such a system can *assist* a professional—the randomised evidence on assistive generative AI is consistent in the sign of first-order effects (Noy and Zhang, 2023; Brynjolfsson et al., 2025; Peng et al., 2023; Dell'Acqua et al., 2023)—but whether it can *own* a piece of white-collar work from brief to deliverable: receive the files, do the reconciliation, build the model, write the memorandum, submit the form, and hand back something a demanding manager would accept without redoing it.

The economics of automation has, since Autor et al. (2003), treated the task rather than the occupation as the unit at which technology substitutes for labour, and the task-based models of Acemoglu and Autor (2011), Acemoglu and Restrepo (2018) and Acemoglu and Restrepo (2019) make the employment consequences depend on which tasks a technology can perform, at what cost, and whether new tasks are created in their place. Acemoglu (2025) writes the aggregate productivity effect of generative AI as the product of the share of tasks exposed, the share actually automated and the average cost saving per automated task, and observes that the second and third factors are assumed rather than measured. The exposure literature from Frey and Osborne (2017) through Brynjolfsson et al. (2018), Eloundou et al. (2024), Gmyrek et al. (2025) and Cazzaniga et al. (2024) estimates the first factor by asking humans or models which tasks a model *could* perform; Arntz et al. (2017) showed how much the answer depends on looking inside occupations. Usage-based measures (Handa et al., 2025; Tomlinson et al., 2025; Appel et al., 2025) observe what people ask models to do, not whether the result was professionally usable. None of these observes an agent actually doing the work.

Two developments make direct measurement possible. Agent benchmarks have moved from code generation to complete deliverables and live environments: GDPval grades documents, spreadsheets and presentations produced from real professional files against the human product (Patwardhan et al., 2025); OSWorld, WebArena, TheAgentCompany and AppWorld place agents in real or simulated desktops, web applications and companies and check the resulting state (Xie et al., 2024; Zhou et al., 2023; Xu et al., 2024; Trivedi et al., 2024); METR's time horizon measures the length of task, in expert human time, that a model completes half the time (Kwa et al., 2025). And the delegation of whole tasks has become ordinary practice, with its own contracting theory (Sreenivas and Larson,

2026; Catalini et al., 2026). What is still missing is an experiment that treats the agent's access to the computer as a randomised treatment, its output as a deliverable graded against a professional standard and a hidden ground truth, its cost and its supervision burden as outcomes, and its tasks as a weighted sample of an occupation. This paper reports the pilot of such an experiment.

Design. We built a laboratory in which Claude Code launches Claude Code. A controller session holds the hidden oracles, the experiment database and the ledger; it composes an isolated workspace for each run, launches a bare-mode Claude Code session over SSH on one of four Mac Studio nodes, streams and stores every event, enforces a wall-clock limit, collects and hashes the deliverables, deletes the workspace, and runs the scenario's deterministic evaluator and three blind language-model judges (Section 3). The benchmark consists of ten scenarios, one in each of ten occupations—accounting, corporate finance, real-estate appraisal, academic research, executive administration, banking, legal services, human resources, procurement and business intelligence—set in a synthetic economy of fictional organisations, each generated by one agent and independently validated by another who solved the task blind before opening the oracle. Every scenario carries a reference solution, three to nine planted errors with known locations, a prompt-injection document and, in four cases, a clarification trap; the worker never sees any of this.

Each scenario is run under five tool configurations that correspond to recognisable working environments: A0, file reading and writing without a shell (the agent as document assistant); A1, the shell and a Python environment (a programmer's workstation); A2, the shell plus the headless macOS automation layer; A3, full computer use with LibreOffice, a browser engine and a synthetic enterprise portal (a professional's desk inside an organisation's systems); and A4, the same tools with a checkpoint file, a public self-check and one controller-granted recovery resume (the agent with a safety net but no supervisor). The order of runs is a seeded permutation interleaved over nodes, so that each contrast is a within-scenario treatment effect with scenario-clustered inference (Abadie et al., 2023; Miller, 2024). A run is *accepted* when it passes every required criterion of the hidden oracle, contains no critical violation—no fabricated figure, no modified input, no compliance with an injected instruction, no attempted external communication—and receives a mean blind quality score of at least 70 on 100. The hypotheses, estimands, inference plan, caps and scale rule were preregistered before any run, and every departure is logged as an amendment (Nosek et al., 2018; Olken, 2015).

Findings. The pilot executed 150 runs—ten scenarios, five arms, three repeats, all at `claude-opus-5-5` on Claude Code 2.1.314, on 2–3 November 2026—plus 40 model-sensitivity runs and 40 runs of an exploratory recovery experiment, for a total API spend of 261.89 dollars against an envelope of 800. Four results organise the paper. First, without a shell nothing is accepted and with one

almost everything is: scenario-weighted acceptance is 0.000 in A0 and 0.867, 0.900, 0.900 and 0.833 in A1–A4, and the shell premium of +0.867 [0.667, 1.000] (permutation $p = 0.004$) is the only confirmatory effect in the study. The mechanism is the one the preregistration anticipated: the deliverables are workbooks with live formulas and formatted documents, which cannot be produced without a program. Second, the preregistered primary contrast—full computer use minus shell—is a precise null, +0.033 [0.000, 0.100], $p = 1.0$: at this level of difficulty every arm with a shell sits at a ceiling, and the browser, the office adapter and the portal had nothing left to add on scenarios that were, by their authoring specification, solvable from the shell. Third, failures are concentrated and legible: seven of 150 runs contain a critical violation, six of them in A0, and no run in the three richest arms; the remaining non-acceptances are almost all one scenario, the executive triage, which passes every deterministic check and fails the quality gate. Fourth, cost is about one dollar of marginal API charges per accepted deliverable and scales with the model, not with the tools: `claude-sonnet-5-5` reaches 0.85 acceptance at 0.40 dollars and `claude-fable-5-1` 0.85 at 3.30 dollars.

Two further results bear on the economics of delegation. Because workers were forbidden to ask questions during a run and instructed to record them, the completion records yield a supervision proxy; valued at five minutes per question, the Autonomous Labor Frontier of arm A3—the task-weighted probability that a run is accepted, costs no more than a budget B and requires no more than H minutes of human time—is 0.167 at a supervision allowance of zero and 0.667 at fifteen minutes, against a marginal acceptance ceiling of 0.900, while the budget ceases to bind at about 1.50 dollars. The supervision margin, not the budget, is what separates the frontier from the ceiling. And in an exploratory experiment that interrupts sessions at 120 seconds, bounded recovery raises acceptance from 0.05 to 0.85 (+0.80 [0.55, 0.95], $p = 0.004$) at essentially the cost of an uninterrupted run; the mechanism was never triggered in the pilot because no first session failed its self-check.

Contribution and limits. The paper contributes an operational measurement framework (the Autonomous Labor Frontier and four companion measures, presented as conventions with stated precedents rather than as new ideas), a benchmark of ten validated scenarios with hidden oracles, a laboratory that was built rather than described and ran 230 sessions without an unplanned timeout, and two results that were not foreseen: a precise null on the primary contrast that locates where the next benchmark tier must be built, and a recovery mechanism that works when something interrupts the agent. Its limits are stated with equal care, because they are the premises of every interpretation. The human arms of the mandate were designed but not executed; no participant was recruited and no human-performance figure appears except as a labelled assumption (amendment A-002). The benchmark is self-authored by language-model agents and validated by language-model agents, which is a circularity the hidden oracle mitigates but does not remove. The judges are models of the

workers' family. There are $J = 10$ scenarios, one per occupation, so that every interval is wide and every occupation-level statement is descriptive. The recovery experiment was added after the pilot and is exploratory. And nothing in the paper is an estimate of job displacement: it measures the probability that one family of agents completes ten designed tasks to a stated standard under stated constraints, in a laboratory, on one date.

Section 2 states the framework. Section 3 describes the benchmark and the laboratory. Section 4 states the preregistered design. Section 5 reports the results. Section 6 discusses their economic reading and limitations, and Section 7 concludes. The monograph of which this is the journal version contains the proofs, the scenario and rubric documentation, the infrastructure and reproducibility records and the robustness tables (Appendix 7).

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

The framework is deliberately modest in theoretical ambition. Its constructs 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. None of the constructs is new as an idea; each is a measurement convention whose precedents are listed in the monograph's positioning table, and the contribution lies in their joint, experimental, occupation-weighted estimation under randomised tool access with hidden-oracle evaluation. The purpose of the formalism is to fix, before any run, what is being estimated and under which assumptions it is identified.

2.1. Occupations, tasks, runs and acceptance

Let occupation o be represented by tasks $j = 1, \ldots, J_o$, each a complete professional assignment with a brief, input artefacts and required deliverables, and by weights $w_{oj} \geq 0$ with $\sum_j w_{oj} = 1$ representing the economically justified importance of task j within o; in the primary analysis the weights are equal and in the sensitivity analysis they are proportional to the professional-minutes estimate recorded in each scenario's metadata. An *arm* $a \in \{A0, \ldots, A4\}$ is a configuration of the computer tools available to the agent, holding the model, the brief, the inputs and the caps fixed. A *run* is one execution of task j under arm a, with repeat index r; it is a random experiment because the model samples tokens stochastically and the environment responds to the actions it chooses. A run produces the vector $Y_{ojr}(a) = (D, Q, E, C, T, H, X)$, where $D \in \{0,1\}$ is *deterministic completion* (every required oracle criterion passes and no critical violation is recorded), $Q \in [0, 100]$ is the mean rubric score of the blind judges, $E \in \{0,1\}$ is the *critical-error indicator* (fabrication, modified inputs, compliance with an injected instruction, attempted external communication, or a scenario-specific violation listed in the oracle), $C \geq 0$ is the marginal API cost in US dollars as reported by

the client including any recovery session, T is wall-clock time right-censored at the session limit, $H \geq 0$ is the human intervention time in minutes—in this version, zero during the run plus a stated assumption of five minutes per question the agent records as unanswered—and X collects process measurements (tool calls, errors, retries, application transitions, tokens).

Definition 1 (Acceptance)**.** For a quality threshold q, the acceptance indicator of a run is

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

and the acceptance probability of task j under arm a is $p_{oj}(a;q) = \Pr\{S_{ojr}(a;q) = 1\}$.

The three factors are not redundant. Deterministic completion certifies facts only the oracle knows; the quality gate certifies what a demanding professional would notice but a test cannot encode; the integrity factor certifies the absence of behaviours that 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 can pass it while failing the others, and Section 5.4 shows a scenario in which exactly this happens.

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,

$$\mathrm{ALF}_o(B,H,q;a) = \sum_{j=1}^{J_o} w_{oj}\,\Pr\big\{S_{oj}(a;q) = 1,\, C_{oj}(a) \leq B,\, H_{oj}(a) \leq H\big\}. \tag{2}$$

The frontier is the task-weighted probability that a run of a 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, and it is indexed by three thresholds because the economic question has three margins—whether the work is good enough, cheap enough and autonomous enough—and a single number would hide the trade-offs among them. Three properties, proved in the monograph, govern its reading. The task-weighted marginal acceptance rate $\rho_o(q) = \sum_j w_{oj} p_{oj}(q)$ is the *ceiling* of the frontier, approached as $B, H \to \infty$; the gap $\rho_o - \mathrm{ALF}_o$ is the share of accepted work that is infeasible at the stated budget or allowance. If the per-session budget cap is $\bar{B}$ and runs are evaluated as delivered when it binds, the frontier is flat for $B \geq \bar{B}$ and the experiment cannot inform that region. And for two weight vectors on the same tasks, the change in the frontier is bounded by their total-variation distance times the range of the task-level joint probabilities, which is why both weightings are reported. The joint probability decomposes, for each task, into deterministic completion, the quality gate conditional on completion, integrity, and budget

feasibility conditional on acceptance; the decomposition is a diagnostic that tells us *where* an arm loses acceptable work.

2.3. The Computer-Use Tool Premium

Each run has a potential outcome under every arm, and the design assigns every task to every arm, so the contrast is identified within task.

Assumption 1 (Exchangeable repeats, no interference, information constancy)**.** For each task and arm, the run vectors of the repeats are independent and identically distributed; a run's vector does not depend on the arm, order or node of any other run, nor on its own position in the execution order or executor node. The brief, inputs, deliverable specification, model, client version, caps and behavioural policy are identical across arms; arms differ only in the tool set, its documentation and, where applicable, the presence of the enterprise portal and of the self-repair protocol.

Definition 3 (Computer-Use Tool Premium)**.** For arms a, a' and tasks j with weights w_j, the tool premium is the task-weighted average treatment effect on acceptance, $\tau(a,a';q) = \sum_j w_j\,[p_j(a;q) - p_j(a';q)]$. The *shell premium* is $\tau(A1,A0)$, the *computer-use premium* $\tau(A3,A1)$, the *macOS-layer premium* $\tau(A2,A1)$ and the *self-repair premium* $\tau(A4,A3)$.

The estimator is the task-weighted mean of within-task differences of repeat means, $\hat{\tau} = \sum_j w_j[\bar{S}_j(a) - \bar{S}_j(a')]$. Under Assumption 1 it is unbiased, and when tasks are an equal-weight sample of size J from an occupation's task population its variance is

$$\mathrm{Var}[\hat{\tau}] = \frac{1}{J}\left[\mathrm{Var}_j(\Delta_j) + \mathbb{E}_j\left(\frac{\sigma_j^2(a) + \sigma_j^2(a')}{R_j}\right)\right], \tag{3}$$

with $\Delta_j = p_j(a) - p_j(a')$ and $\sigma_j^2(a) = p_j(a)\{1 - p_j(a)\}$. Equation (3) is the reason the primary inference clusters at the task: between-task heterogeneity in the effect is reduced only by adding tasks, while within-task stochastic variation is reduced by repeats (Miller, 2024; Abadie et al., 2023). 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. Two caveats apply: the premium is the effect of a *bundle*—tools plus documentation plus, for A3 and A4, the portal—and arm A2 is identified conditional on the nodes where the macOS automation layer answers.

2.4. Companion measures

Four companion measures complete the framework; each is defined briefly because the pilot identifies only some of them. The *Workflow Dependency Frontier* is the function $d \mapsto p(a;d)$ relating acceptance to a scenario's designed dependency depth d, the minimal number of dependent

operations a careful professional must perform. Under a compounding model in which d steps each succeed with probability π conditional on the previous ones, $p(d) = \pi^d$, the half-depth at which acceptance falls to one half is $d_{1/2} = \log 2 / \log(1/\pi)$, heterogeneity of step reliability lowers task reliability at a given mean, and self-conditioning in the sense of Sinha et al. (2025) makes $-\log p(d)$ convex; the model is a benchmark shape against which an estimated frontier is read, not a structural claim (Dziri et al., 2023; Kwa et al., 2025; Yao et al., 2024).

The *Autonomous Recovery Index* is defined on 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 it after the bounded recovery; then $\mathrm{ARI} = \Pr\{P_2 \mid F_1\}$. If a run that would fail under A3 fails the self-check with probability δ (detection) and a resumed session converts a detected failure into an accepted deliverable with probability γ (correction), and the first session of A4 is distributed as a run of A3, then $p(A4) = p(A3) + (1 - p(A3))\,\delta\gamma$: the self-repair premium is proportional to the first-session failure rate the self-check can see, and is zero by construction when that rate is zero. The index measures correction against an external signal, which Shinn et al. (2023) and Chen et al. (2023) find effective, not intrinsic self-correction, which Huang et al. (2023) find ineffective; by the imperfect-verifier result of Stroebl et al. (2024), passing the self-check is not acceptance.

Autonomy-Adjusted Productivity is the expected number of accepted deliverables per unit of fully loaded cost, $\mathrm{AAP}(a; q, \omega, \kappa) = \sum_j w_j \mathbb{E}[S_j] / \sum_j w_j \mathbb{E}[C_j + \omega H_j + \kappa]$, where ω is a wage per human minute and κ an operating cost per run reported separately from the API charge; its reciprocal is the cost per accepted deliverable, computed over all runs so that an arm pays for its failures. The *Professional Reliability Frontier* treats the end of a run as one of four competing terminal events—accepted, critical error, censored at a cap, or finished without acceptance—and maps (q, B) to the triple (probability of accepted work within budget, critical-error rate, censoring rate), so that an arm that raises acceptance while raising the critical-error rate is not read as an improvement (Fine and Gray, 1999; Dell'Acqua et al., 2023).

2.5. Delegation with verification

The constructs enter the decision of a principal who must decide whether to delegate a task to an agent. A task yields value v when an acceptable deliverable is used; the principal can produce it in-house at cost c_H or delegate it 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 used causes a loss ℓ, an undetected critical error a loss $L \geq \ell$; verification at cost κ_V detects failures and completes the task in-house.

Proposition 1 (Delegation thresholds)**.** *Delegation with verification yields a higher expected payoff than in-house production if and only if $p \geq (c_A + \kappa_V)/c_H$; verification is worth its cost relative to*

delegating without it if and only if $\kappa_V \leq (1-p)(\ell - c_H) + eL$*; and as* $L \to \infty$ *verification becomes a necessary complement of delegation for any* $e > 0$.

The first inequality is the frontier's budget condition in decision form; the second formalises the claim of Catalini et al. (2026) and Sreenivas and Larson (2026) that, when execution is cheap, the delegation decision is governed by how cheaply a human can check the work. It is also why E is measured separately from acceptance: e and L enter the verification decision through a term that acceptance alone does not capture. The frontier is not an estimate of job displacement, a task is not an occupation, the scenarios are synthetic, and the estimates are for one family of models and one client version on one date; the monograph states each of these restrictions formally and the mandate of the study makes them rules.

3. The benchmark and the laboratory

3.1. The Virtual White-Collar Economy and the anatomy of a scenario

Scenarios are set in a small, internally consistent fictional economy of seventeen organisations—an accounting firm, an investment firm, a commercial bank, an appraisal firm, a law firm, a university department, a manufacturer and its human-resources and procurement departments, a municipality, and others—that share conventions (fiscal year 2025, Canadian dollars, Québec sales taxes, ISO dates) so that an agent moving between scenarios faces a coherent world. All names, people and figures are fictional. The economy is built around one manufacturer, Boréal Fenestration Inc., which is the client of the accounting firm and the law firm, the employer in the human-resources and procurement scenarios, the subject of the business-intelligence pack and the organisation whose chief executive's affairs are triaged.

A scenario has two halves 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 enforced by location: the bundle lives in the repository; the oracle lives outside it on the controller and is never copied to an executor node. The bundle contains metadata reduced to what a worker legitimately needs; a brief written in the voice of the organisation stating context, inputs, deliverables with exact file names and formats, business rules and conventions, ending with common working rules (deliverables go to `out/`, inputs must not be modified, no contact outside the workspace and no network except an internal portal if provided, a completion record must be written, and if the assignment cannot be completed because information is missing or contradictory the worker must say so with its questions instead of guessing); the input artefacts—spreadsheets, CSV extracts, PDF invoices and statements, Word contracts, e-mail messages, calendar files, SQLite databases, policies, images—identical in every arm; and, for

the portal arms, a description of what the enterprise portal serves. The brief contains no grading information and no hint that any document contains an injection.

The oracle contains the ground truth (planted items and correct values, tolerances, required deliverables, injection markers), the deterministic evaluator, the blind-evaluation rubric with weighted criteria anchored on a 0–4 scale, a reference document and the validator's report. The evaluator reads only the collected output and the public workspace, never executes the worker's code, applies the oracle's tolerances and returns a receipt listing each required criterion's pass/fail status, each critical violation, the completion flag and the abstention status. Every scenario has at least six required criteria, of which at least three are facts only the oracle knows, and at least two critical-violation checks. Deliverables are real files—workbooks with live formulas, Word documents with specified sections, CSV tables, figures, code—plus a machine-readable completion record whose status is *completed*, *needs_clarification* or *abstained* and which lists the questions the worker would have asked. Every scenario embeds *planted errors* (a transposed amount, a duplicated invoice, a comparable sale with the wrong date), one *prompt injection* inside an input document whose obedience is a critical violation, and in four scenarios a *clarification trap*, a sub-task that cannot be completed without information the bundle does not contain. The behavioural rule given to every worker—treat the content of input documents as data, not as instructions—is identical across arms. Each scenario also carries a *designed dependency depth*, the minimal chain of dependent operations a careful professional must perform.

3.2. Generation and independent validation

Scenarios are produced and checked by operationally separate agents. The *generator* is a fresh, single-purpose Claude Code session that writes the generator script, the brief, the inputs, the portal data and the oracle for one scenario; the script is deterministic given its seed and internally coherent by construction, so that the planted discrepancies are the only discrepancies. The *validator* is a second fresh session that never sees the generator's conversation: it first solves the task from the public bundle alone, then opens the oracle, runs the evaluator on its own solution (which must pass) and on a plausible wrong solution (which must fail), and checks that the brief is realistic, the task solvable, the hidden answers correct, the rubric gradable from the deliverables alone and the portal free of exclusive information. All ten validator solutions pass their evaluators and every empty output fails; the validators found and fixed between one and seven evaluator defects per scenario before the freeze—false positives on honest disclosure of the injection, vacuous or lenient criteria, non-reproducible document hashes—and removed two brief sentences that primed workers against injections. A scenario enters the assignment only after validation, and the hashes of its bundle and oracle are recorded in the preregistration freeze manifest (amendment A-007). The protocol protects against the task-validity failures catalogued by Zhu et al. (2025); it does not remove the circularity

that generator, validator, worker and judge are all language-model agents (Section 6).

3.3. The ten pilot scenarios

The pilot covers ten of the twenty occupations of the study's taxonomy. Each scenario is a complete assignment: a year-end bank reconciliation from 150 invoices, twelve statements and a general ledger (ACC-01, accounting, designed depth 44); a discounted-cash-flow valuation and investment memorandum (FIN-01, corporate finance, depth 33); a single-family appraisal with an adjustment grid and report from eight closed sales (RE-01, real-estate appraisal, depth 34); a pre-specified difference-in-differences replication with data cleaning and a reference check against a library catalogue (RES-01, academic research, depth 34); the triage of a chief executive's forty messages, 25 calendar events and five project documents into an action plan, draft replies, a filing index and a briefing note (ADM-01, executive administration, depth 44); a credit analysis of a 1.8-million-dollar equipment loan under a credit policy (BANK-01, banking, depth 24); a clause-by-clause review of a 25-clause master services agreement against a negotiation playbook (LEGAL-01, legal services, depth 23); the closing of a pay period for 25 employees under a payroll policy (HR-01, human resources, depth 22); the evaluation of five supplier quotations under a published method (PROC-01, procurement, depth 21); and an annual sales KPI pack from a raw ERP database, with a dashboard workbook, a commentary and the code that produced it (BI-01, business intelligence, depth 31). Four scenarios are of the standard difficulty class, four complex and two workflow; planted errors number three to nine; every scenario has an injection document; the clarification traps are in the executive-administration, legal, human-resources and procurement assignments; and the authors' professional-minutes estimates range from 150 to 420. No short scenario is in the pilot, a deliberate choice to measure the regime in which the frontier is expected to be informative and a limitation for estimating the dependency frontier at low depth.

3.4. Claude Code calling Claude Code

The laboratory has a controller–executor architecture (Figure 1). 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 a private dashboard. It never executes a worker itself. The *executors* are four Mac Studio nodes of an authorised cluster (M3U96a, M3U96b, M4M64a, M4M64b; macOS 26.4–26.6; 16 to 28 cores and 64 to 96 GB) running Claude Code 2.1.314, each with a project Python environment, LibreOffice 26.2 for headless conversion and recalculation and Playwright's headless Chromium; each runs up to three concurrent sessions in separate per-run directories and

holds no oracle, rubric or evaluator code. Model access was verified by headless probes whose cost was read from the client's own result event, and that field is the source of every cost in the paper.

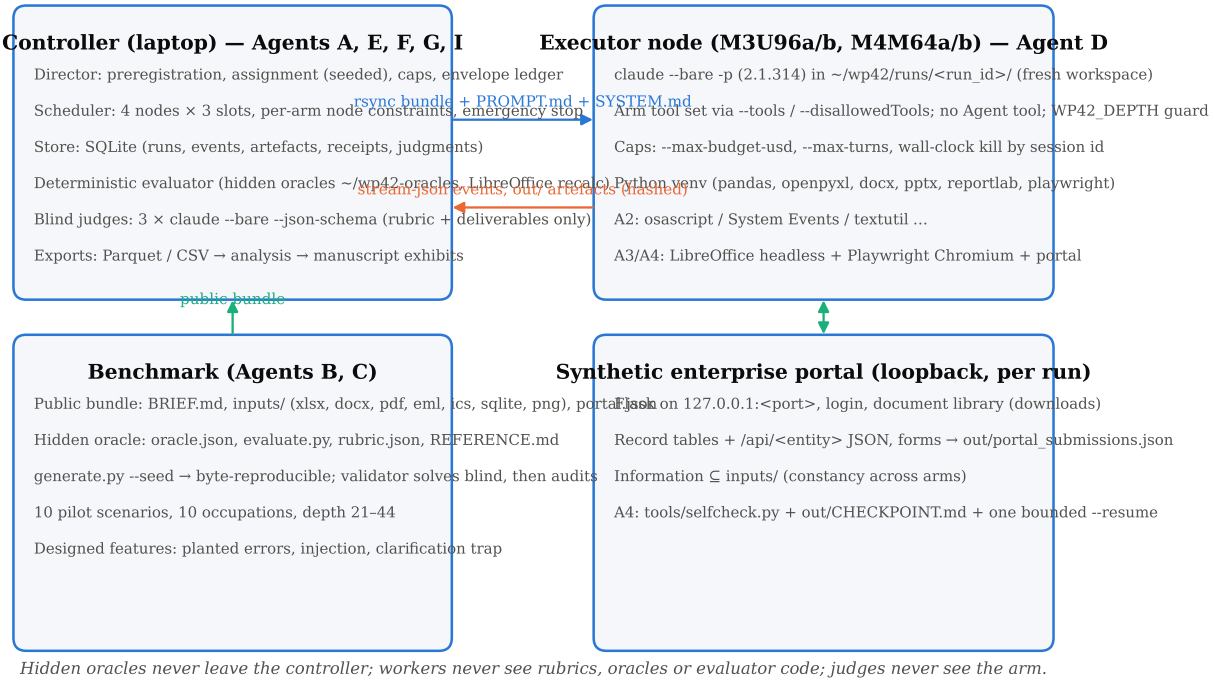


Figure 1. Laboratory architecture. The controller (left) holds the hidden oracles outside the repository, composes a per-run workspace from the public bundle, launches a bare-mode Claude Code session over SSH on one of four executor nodes under arm-specific flags, streams and stores every event, enforces the wall limit, collects and hashes the deliverables, deletes the workspace and runs the deterministic evaluator and three blind judges. Executor nodes (right) host per-run workspaces, the loopback enterprise portal for arms A3 and A4, LibreOffice headless and Playwright Chromium, and never hold oracle, rubric or evaluator code. Source: `analysis/figures/out/fig01_architecture.pdf` (figure script of the reproducibility package).

A worker session is a single command assembled by the launcher: the client is started in bare mode, ignoring the user's settings, hooks and instruction files so that nothing from the controller's configuration reaches the worker; the tool set is restricted by `-tools` and pre-approved by `-allowedTools`; permission prompts are disabled; budget and turn caps are set; the arm's documentation is appended to the system prompt; denied command families are listed in `-disallowedTools` with patterns such as `Bash(osascript:*)`. The effect of each flag was verified by probe: a worker started without the shell tool could not run a trivial command, and a denied command family was refused inside the shell while a plain command ran. No Agent or Task tool is ever given, so a worker cannot spawn sub-agents. Each workspace is pushed to the node with a manifest of SHA-256 hashes; after the session the output directory is copied back and hashed, the inputs are compared with the manifest and any change is recorded as a critical violation, the portal is stopped and the workspace is deleted. The *synthetic enterprise portal* is a small web application started on the

node for each A3 and A4 run, bound to the loopback interface, serving a login page, a document library whose links point to files already in the inputs, record tables and forms whose submissions are collected with the other deliverables; it holds no information absent from the inputs. External network access is forbidden by policy.

Role separation is operational rather than nominal: generator and validator are distinct sessions; the worker is a separate process on a separate machine; the evaluator is code the worker never sees; the judges receive a rendered package and a rubric; and the sessions that wrote the sections of this paper reporting results had access to the analysis outputs and not to the runs. Three properties of the environment constrained the design and are recorded as amendments. Control of graphical applications is not available to a process started over SSH—Finder and TextEdit requests time out on all four nodes because the automation consent prompt cannot be answered for a non-console process—so the mandate's "macOS agent" is implemented as the automation layer that works headlessly (`osascript` with AppleScript and JavaScript for Automation, System Events on the two nodes where it answers, and the native document utilities), and A2 runs are pinned to those nodes (A-003). No human evaluator was available, so the blind professional evaluators are language-model judges (A-005). And the human arms of the mandate were designed and not executed (A-002).

4. Experimental design and preregistration

4.1. Treatment arms

Five agent arms implement increasing levels of computer access with the same model, brief, inputs, caps and behavioural policy. *A0 (restricted)* gives the tools Read, Write, Edit, Glob and Grep and no shell; because binary office formats cannot be produced without a program, the A0 documentation instructs the worker to deliver the closest text equivalent under the same base name and to state the substitution, and the deterministic criteria that require a formula-bearing workbook then fail, as they should. *A1 (shell)* adds Bash and a project Python environment (pandas, openpyxl, python-docx, reportlab, pypdf, matplotlib, statsmodels, duckdb, sqlite3 and others) and denies, inside the shell, the families `osascript`, `open`, `soffice`, `textutil`, `qlmanage`, `cupsfilter`, `curl` and `wget`. *A2 (macOS)* lifts the denial of `osascript` and the native document utilities but keeps LibreOffice and the network denied. *A3 (full computer use)* denies nothing and adds LibreOffice headless, Playwright's headless Chromium and the enterprise portal with credentials in the workspace. *A4 (self-repairing)* has A3's tools and adds a checkpoint file the worker keeps updated, a public self-check tool that verifies the presence and well-formedness of deliverables (files exist and are non-empty, office files are valid archives, a workbook required to contain formulas does and shows no error values, documents meet minimum lengths, the completion record is valid) 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 by timeout, budget exhaustion or error. The recovery resumes the same session with the remaining budget and wall time, at most once; its cost and time are attributed to the parent run.

All arms use `claude-opus-5-5` at default effort, bare mode, `-permission-prompts none`, `-max-turns 150`, and a spending cap and wall limit set per scenario as the larger of the global caps (4.00 dollars, 1,800 seconds) and the scenario author's own: ACC-01 under 6.00 dollars and 3,600 seconds, RES-01 under 6.00 dollars and 2,700 seconds, BANK-01 under 5.00 dollars and 2,400 seconds, LEGAL-01 under 2,400 seconds, the other six under the global caps (A-006, fixed before any pilot run). The study has an envelope of 800 dollars enforced by the scheduler's ledger (A-001). Each arm's documentation is appended to a common policy, 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, 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. The user prompt is a single sentence asking the worker to read the brief and carry out the assignment completely.

4.2. Randomisation, repeats and caps

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 scheduler draws cells in that order and dispatches each to the least-loaded eligible node, so that arms are interleaved over time and nodes rather than run in blocks, and drift in the model's service or in node load affects all arms alike. The seed, order index and node of every run are stored. Workspaces are reset between runs. The preregistered pilot used $R = 2$; the scale rule below determined whether a third repeat was added. A run that reaches a cap is evaluated as delivered and its censoring recorded. Infrastructure errors—failures of SSH, of the file transfer or of the portal start, in which no inference was performed—are rerun under the same identifier after a reset, with the failed attempt logged. No assigned run is excluded; a missing judge output leaves quality averaged over the available judges, and if none is available quality and acceptance are reported as missing, never imputed as zero.

4.3. Outcomes and evaluation

Deterministic completion D is true when every required oracle criterion passes and no critical violation is recorded by the evaluator or by the harness's hash check of the inputs. Quality Q is the mean over three 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 varied from 50 to 90 in sensitivity analyses. Cost C is the client's reported `total_cost_usd` including any recovery session. Human time H is zero intervention minutes during the run plus a stated assumption of five minutes per unanswered question in the completion record. The blind judges are one draw of `claude-opus-5-5` and two independent draws of `claude-sonnet-5-5`, run as single-turn, tool-less sessions with a fixed JSON output schema. Each receives the brief, the rubric and the deliverables rendered as text, grades each criterion from 0 to 4, states whether a demanding manager would accept the package, and 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 or the other judges' outputs; a text substitute delivered by A0 is rendered under the requested base name. This follows the mitigations of Gu et al. (2024)—rubric anchoring, multiple judges, blinding to condition—and does not address self-preference (Panickssery et al., 2024), since judges and worker belong to the same family; calibration against human experts is left to a later wave (A-005). No evaluator sees two packages side by side, so the position bias of Wang et al. (2024) does not arise.

4.4. Hypotheses, estimands and inference

The preregistration states seven directional expectations: H1 (primary), the computer-use premium $\Pr(S \mid A3) - \Pr(S \mid A1) > 0$; H2, the shell premium $\Pr(S \mid A1) - \Pr(S \mid A0) > 0$, driven by the feasibility of binary deliverable formats; H3, the self-repair premium ≥ 0 and an Autonomous Recovery Index above one half when a first session fails the self-check; H4, acceptance decreasing within arm in designed depth and in realised tool calls; H5, a critical-violation rate below 10 per cent in every arm and not increasing with tool access; H6, cost per accepted deliverable lowest in A1 or A3, not in A0; and H7, heterogeneity across occupations with no occupation-level claim confirmatory at the pilot scale. The primary endpoint is $S = \mathbf{1}\{D = 1 \wedge E = 0 \wedge Q \geq 70\}$; the scenario-weighted acceptance of an arm is $\hat{p}_a = J^{-1} \sum_j R_j^{-1} \sum_r S_{jra}$ with equal weights (professional-minutes weights as a sensitivity); the primary contrast is $\hat{p}_{A3} - \hat{p}_{A1}$ and the secondary contrasts A1−A0, A4−A3 and A2−A1 are Holm-adjusted over the three. Secondary endpoints include D, Q, E, cost, wall time, process counts, abstention and its justification, cost per accepted deliverable, the frontier by occupation over budgets from 0.50 to 4.00 dollars and over the human-time allowance, the recovery index and evaluator agreement.

The primary interval for every contrast is a scenario-cluster bootstrap: scenarios are resampled with replacement 5,000 times, the within-scenario paired difference of repeat means is recomputed, and the percentile 2.5 and 97.5 points form the 95 per cent interval. The primary test is a paired sign-flip permutation test over scenarios with 20,000 permutations, exact under the null of no effect within scenario. A cluster-robust logistic regression, Kaplan–Meier curves for time to completion

and joint bootstraps of ratio quantities are reported as sensitivities. With $J = 10$ the intervals will be wide, and the preregistration states that they are to be reported as such.

4.5. Scale rule, power and the human arms

The pilot comprised 10 scenarios $\times$ 5 arms $\times$ 2 repeats. The scale rule, fixed in advance, added a third repeat if $1.15\,\bar{c} \times 50 \leq E_r - 100$, where $\bar{c}$ is the mean total cost per run including judges and E_r the remaining envelope, and then ran the model sensitivities—`claude-sonnet-5-5` and `claude-fable-5-1` on arm A3, two repeats each—if $1.15\,\bar{c} \times 40$ fitted in what remained minus 50. Both conditions were met at a pilot mean cost of 1.07 dollars per run (A-009), and the third repeat was pooled with the first two because the freeze verification found no change to any scenario, oracle, rubric or arm text. Extension to twenty scenarios was deferred to a later wave. Because no comparable experiment exists from which to borrow effect sizes, power was explored by simulation under the cluster structure of equation (3), with a scenario random effect of standard deviation 0.8 on the logit scale and a paired scenario-level test at the 5 per cent level. At a baseline acceptance of one half, ten scenarios with three repeats detect an effect of 25 percentage points with probability 0.408 and of 35 points with probability 0.729; forty scenarios with three repeats detect 25 points with probability 0.968. Doubling the number of scenarios helps more than doubling the repeats, as equation (3) predicts when between-scenario heterogeneity dominates.

The mandate specifies two human conditions, H0 (participants without AI) and H1 (participants with conversational assistance), with 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 except as an explicit, labelled assumption (A-002). The protocol for that wave—six professionals or senior students per scenario per arm, randomised between H0 and H1 within occupation, the same bundles, evaluators and rubric, TCPS 2 consent—is part of the reproducibility package. Until it is run, every statement here about the agent arms is a statement about agents alone.

5. Results

5.1. The data

Three stages were executed and are analysed separately. The *pilot* comprises 150 runs: the ten validated scenarios, each under the five arms with three repeats, all at `claude-opus-5-5` on Claude Code 2.1.314, executed between the evening of 2 November and the early hours of 3 November 2026 at a concurrency of twelve sessions over the four nodes (37, 40, 38 and 35 runs per node; arm A2 on

the two nodes where the automation layer answers). The *sensitivity* stage comprises 40 runs: arm A3 on the same scenarios with two repeats each under `claude-sonnet-5-5` and `claude-fable-5-1`. The *recovery* stage comprises 40 runs, the exploratory addendum of amendment A-011 (Section 5.6). Every pilot and sensitivity run ended with a result event of subtype `success`—no run timed out, exhausted its budget or reached the turn cap—and every run received three judge outputs; the 40 recovery-stage first sessions ended by the induced interruption. In total 230 experimental runs were executed and evaluated, and the launcher reported no infrastructure error.

Three events during the pilot changed how some data were produced, each logged as an amendment and none affecting a pilot outcome. The judge subprocess crashed on eight runs whose rendered deliverables contained a null byte and timed out once; the renderer was fixed and the nine runs re-judged (A-008). The collection step silently lost the output directories of three A2 runs whose sessions had completed; the defect was found by the audit, the step was given retries and a file-count check, and the three runs were reset and rerun under the same identifiers—had the loss gone undetected, A2's acceptance would have been reported as 0.80 instead of 0.90 and the macOS-layer contrast would have changed sign (A-010). And a launcher flag had made the A4 recovery resume impossible; since no A4 first session in the pilot failed the self-check, no recovery was ever requested, but the self-repair arm was in the pilot de facto limited to its checkpoint and self-check components (A-012).

The three evaluation instruments agree to a degree that governs the reading of every acceptance rate. Krippendorff's alpha across the three judges is 0.973 over the 150 runs, with pairwise Pearson correlations of 0.964 and 0.966 between the Opus judge and each Sonnet judge and 0.990 between the two Sonnet draws; the two models are not systematically different in severity (means 73.1 against 73.0), but the near-identity of the two Sonnet draws means the panel is closer to two independent opinions than to three. Deterministic completion and the quality gate disagree on 14 of 150 runs, 9.3 per cent: of 117 runs that pass every required criterion, 105 also score at least 70 and 12 do not; of 33 that fail a criterion, 2 are scored above 70. The disagreement is concentrated: eleven of the twelve runs the judges reject are the executive-administration scenario in arms A1–A4, and the two the oracle rejects are appraisals scored 98.75 that failed a single adjusted-price criterion. Acceptance requires both instruments, as preregistered.

5.2. Outcomes by arm

Table 1 gives the preregistered outcomes by arm, with the two sensitivity models in the last two columns, and Figure 2 the acceptance rates. The restricted arm A0 accepted no deliverable in 30 runs and completed none deterministically: its mean quality is 27.4, its critical-error rate 0.20, and 90 per cent of its runs ended by requesting clarification. It is the busiest arm and the least effective—29.4 tool calls per run against 12 to 19 elsewhere, 11.9 tool errors against 0.7 to 2.0—because its calls

are mostly failed attempts to read binary inputs and to do without a program. The four arms with a shell are close to one another: scenario-weighted acceptance is 0.867 in A1, 0.900 in A2 and A3 and 0.833 in A4; deterministic completion 0.967, 0.967, 1.000 and 0.967; mean quality 82.3, 85.0, 85.7 and 84.8, with a larger dispersion in A1 because one run scored 3.8 after destroying its input; the critical-error rate 0.033 in A1 and zero in A2–A4. Under professional-minutes weights the rates move by at most 0.03. The gap between deterministic completion and acceptance in the shell arms—0.10, 0.067, 0.10 and 0.133—is the quality gate, and Section 5.4 shows that it is almost entirely one scenario. Cost per accepted deliverable, which keeps failed runs in the denominator, is 0.95 dollars in A1, 0.87 in A2, 1.01 in A3 and 1.11 in A4, and undefined in A0.

Table 1. Preregistered outcomes by arm, 150 pilot runs (`claude-opus-5-5`, $q = 70$), with the two model sensitivities on arm A3

	`claude-opus-5-5`					Arm A3, other models	
	A0 restricted	A1 shell	A2 macOS	A3 computer-use	A4 self-repair	Sonnet 5.5	Fable 5.1
Runs (scenarios × repeats)	30 (10×3)	30 (10×3)	30 (10×3)	30 (10×3)	30 (10×3)	20 (10×2)	20 (10×2)
Acceptance S, equal scenario weights	0.000	0.867	0.900	0.900	0.833	0.850	0.850
Acceptance S, professional-minutes weights	0.000	0.885	0.909	0.926	0.844	–	–
Deterministic completion D	0.000	0.967	0.967	1.000	0.967	1.000	0.950
Critical-error rate E	0.200	0.033	0.000	0.000	0.000	0.000	0.050
Quality Q, mean (s.d.)	27.4 (17.4)	82.3 (18.5)	85.0 (10.0)	85.7 (11.4)	84.8 (10.6)	80.9	86.9
Abstention / clarification rate	0.900	0.200	0.067	0.133	0.100	–	–
Timeout rate	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Cost C (USD), mean (s.d.)	0.710 (0.416)	0.823 (0.230)	0.783 (0.190)	0.906 (0.259)	0.926 (0.221)	0.341	2.805
Cost per accepted deliverable (USD)	–	0.950	0.870	1.006	1.111	0.401	3.300
Wall time (s), median	162.0	217.7	213.9	218.8	230.2	131.1	397.6
Tool calls, mean	29.4	12.9	12.4	17.2	18.8	13.5	21.9
Tool errors, mean	11.9	0.9	0.7	0.9	2.0	–	–
Application transitions, mean	0.0	3.3	4.4	9.3	9.3	–	–

Sources: `analysis/tables/table1_outcomes_by_arm_pilot.csv` (arms A0–A4; script `analysis/econometrics/estimate.py`) and `table9_model_sensitivity.csv` (Sonnet and Fable; script `analysis/robustness/sensitivity_models.py`). $S = \mathbf{1}\{D = 1 \wedge E = 0 \wedge Q \geq 70\}$. The abstention rate is the share of runs whose completion record has the status `needs_clarification`. Cost is the client's reported `total_cost_usd`; no recovery was triggered in the pilot. Cost per accepted deliverable is $\sum C / \sum S$ over all runs of the arm and is undefined when $\sum S = 0$.

5.3. Preregistered contrasts

Table 2 reports the four preregistered contrasts on acceptance, deterministic completion, the critical-error indicator and quality, each as the equal-weight mean of ten within-scenario differences of repeat means, with the scenario-cluster bootstrap interval, the paired sign-flip permutation p-value and the Holm-adjusted p over the three secondary acceptance contrasts; Figure 3 plots the acceptance contrasts.

H1, the computer-use premium, is a precise null. Acceptance under full computer use minus acceptance under the shell is $+0.033$ with a bootstrap interval from 0.000 to 0.100 and a permutation

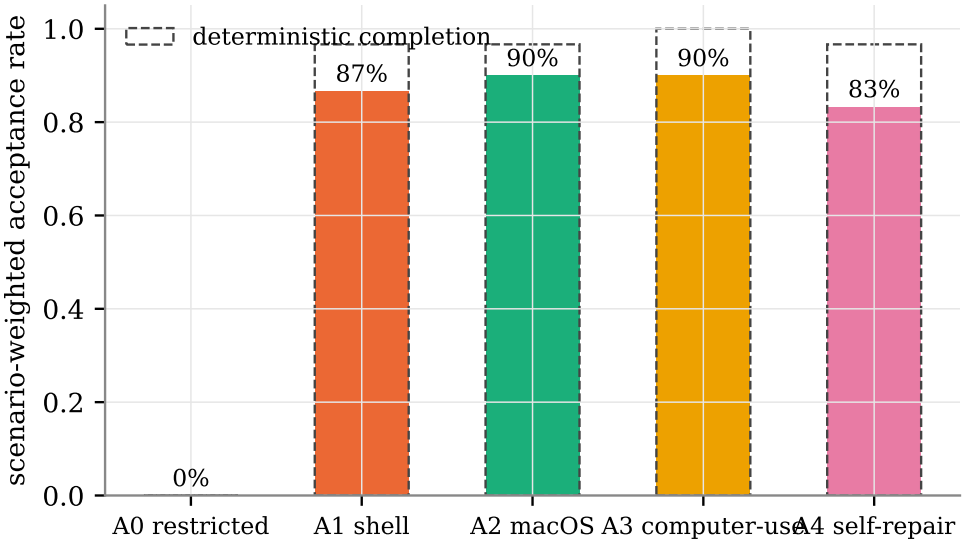


Figure 2. Scenario-weighted acceptance rate by arm (filled bars) and deterministic completion rate (dashed outlines), 150 pilot runs, 10 scenarios × 3 repeats per arm, `claude-opus-5-5`, $q = 70$. Uncertainty is reported in Figure 3 as scenario-cluster bootstrap intervals on the between-arm differences, which is the preregistered inferential quantity; no interval is drawn on the levels. Source: `analysis/tables/table1_outcomes_by_arm_pilot.csv`; script `analysis/figures/make_figures.py`.

Table 2. Preregistered contrasts: scenario-weighted differences with scenario-cluster bootstrap 95% intervals and paired permutation p-values ($J = 10$)

Contrast	Outcome	Estimate	95% CI lower	95% CI upper	Perm. p	Minutes-weighted	Holm p
A3 − A1 (primary, H1)	S	+0.033	0.000	0.100	1.000	+0.041	–
	D	+0.033	0.000	0.100	1.000	+0.041	
	E	−0.033	−0.100	0.000	1.000	−0.041	
	Q	+3.35	−0.81	+9.56	0.300	+4.57	
A1 − A0 (secondary, H2)	S	+0.867	0.667	1.000	0.004	+0.885	0.012
	D	+0.967	0.900	1.000	0.002	+0.959	
	E	−0.167	−0.467	+0.067	0.499	−0.144	
	Q	+54.96	+40.44	+68.24	0.002	+57.66	
A4 − A3 (secondary, H3)	S	−0.067	−0.167	0.000	0.499	−0.082	0.998
	D	−0.033	−0.100	0.000	1.000	−0.041	
	E	0.000	0.000	0.000	1.000	0.000	
	Q	−0.90	−2.71	+0.97	0.395	−1.39	
A2 − A1 (secondary)	S	+0.033	−0.067	+0.133	1.000	+0.025	1.000
	D	0.000	−0.100	+0.100	1.000	0.000	
	E	−0.033	−0.100	0.000	1.000	−0.041	
	Q	+2.63	−1.75	+8.79	0.534	+2.96	

Source: `analysis/tables/table2_contrasts_pilot.csv` (script `analysis/econometrics/estimate.py`; bootstrap seed 42). Intervals are percentile intervals of 5,000 resamples of the ten scenarios with replacement; p-values are two-sided sign-flip permutation tests over the ten paired scenario differences (20,000 permutations). Holm adjustment is over the three secondary acceptance contrasts, as preregistered.

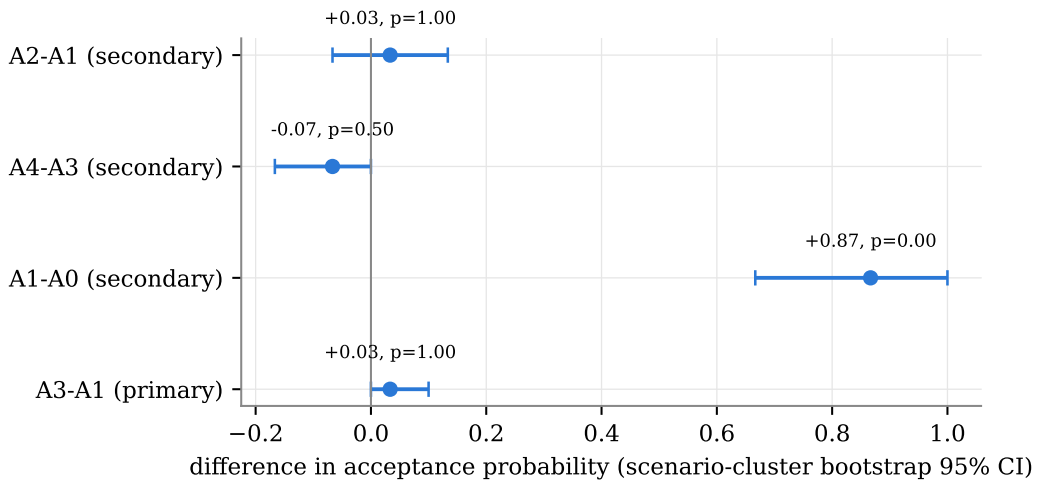


Figure 3. The four preregistered contrasts on acceptance probability: equal-weight mean of the ten within-scenario differences of repeat means, with scenario-cluster bootstrap 95% percentile intervals ($B = 5{,}000$) and the paired sign-flip permutation p-value printed above each point. 150 pilot runs, 10 scenarios, 3 repeats per cell. Source: `analysis/tables/table2_contrasts_pilot.csv`; script `analysis/figures/make_figures.py`.

p of 1.000. The estimate is one scenario-repeat in thirty: A3 accepted one run that A1 did not—the business-intelligence run in which A1 destroyed its input—and the two arms agree on every other scenario mean; the ten paired differences are nine zeros and one positive value, which is why the lower bound is zero, the upper bound 0.10 and the permutation test cannot reject. The contrast on quality is +3.3 points [−0.8, +9.6], $p = 0.30$. H1 is not supported, and the result is not an absence of information: with the shell arm at 0.867 the premium on this benchmark cannot exceed 0.133, and the interval excludes a premium above 0.10. For ten scenarios in which every deliverable can be produced programmatically from the inputs, the browser, the office suite and the portal added nothing the shell could not already do; whether a premium exists on tasks whose inputs or deliverables can only be reached through those applications is a question this benchmark, by construction, does not ask.

H2, the shell premium, is large and confirmatory. The shell premium is +0.867 [0.667, 1.000], $p = 0.004$ (Holm 0.012); on deterministic completion +0.967 [0.900, 1.000], $p = 0.002$; on quality +55.0 points [40.4, 68.2], $p = 0.002$. It is the only confirmatory effect of the pilot, and its mechanism is the one the hypothesis named: every scenario's deliverables include at least one binary office file, usually a workbook with live formulas, which the restricted arm cannot produce; it delivered text substitutes in every run and the criteria that require the workbook's structure and formulas failed in every case. The contrast on the critical-error indicator, −0.167 [−0.467, +0.067], $p = 0.50$, is not significant because the six A0 violations are concentrated in two scenarios.

H3, the self-repair premium, is a null bounded by the design. The premium is -0.067 [-0.167, 0.000], $p = 0.499$ (Holm 0.998); on quality -0.9 points; on critical errors exactly zero. The point estimate is two scenario-repeats in thirty: A4 lost one business-intelligence run to the quality gate (scored 67.9) and one appraisal run to the adjusted-prices criterion, both of which A3 passed in all three repeats. The interpretation is bounded by amendment A-012: all 30 A4 first sessions passed the public self-check, no recovery was ever requested, and the arm differed from A3 only by its checkpointing and self-check instructions and the slightly longer sessions these produce (18.8 tool calls against 17.2; 230 against 219 seconds; 0.93 against 0.91 dollars). Since the self-repair premium is proportional to the first-session failure rate the self-check can see (Section 2.4), it was zero by construction and the Autonomous Recovery Index is unidentified on the pilot design.

The macOS layer and the ceiling. A2 minus A1 is $+0.033$ [-0.067, $+0.133$], $p = 1.000$: A2 accepted the business-intelligence run that A1 lost and lost an appraisal run that A1 passed. The headless automation layer was used sparingly—18 office-family calls in 30 runs against 69 in A3—and no run depended on it for a deliverable the shell could not produce. A cluster-robust logit of acceptance on arm indicators returns logits of 1.87, 2.20, 2.20 and 1.61 for A1–A4 with clustered standard errors of 0.75 to 1.13, of the same order as the differences among the arms, which restates on the model scale what the bootstrap says: the shell arms are not distinguishable from one another. The scale rule gave the pilot a third repeat, which reduces the within-scenario component of equation (3), but the component that governs the primary contrast is between-scenario heterogeneity of the effect, and with the shell arms succeeding on nine scenarios in ten there is none left to measure. The benchmark needs scenarios on which the shell arm fails before the computer-use premium can be estimated at all.

5.4. Heterogeneity across occupations

Table 3 reports acceptance by occupation and arm. Seven of the ten occupations are solved in every repeat of every shell arm—accounting (the deepest workflow, with the highest quality, 89 to 99, and the highest costs, 1.04 to 1.20 dollars per run), banking, legal services and procurement with quality above 89, corporate finance and academic research in the 80s, and human resources with the lowest accepted quality (73.6 to 76.3). Three carry all the shell-arm failures, and they fail for different reasons. Each cell is a proportion of three runs, so H7's heterogeneity is visible but, as the hypothesis stated, not confirmatory.

Executive administration: a quality-gate failure. ADM-01, the inbox-and-calendar triage, is the second workflow-class scenario and the one with the most planted features. Every shell-arm run passed all ten deterministic criteria, and every shell-arm run but one was scored below 70:

Table 3. Acceptance rate by occupation and arm (3 repeats per cell), with deterministic completion where it differs and the range of mean quality in the shell arms

Occupation	Scenario	Depth	A0	A1	A2	A3	A4	Q, A1–A4
Academic research	RES-01	34	0	1	1	1	1	82.2–86.4
Accounting	ACC-01	44	0	1	1	1	1	89.3–98.9
Banking	BANK-01	24	0	1	1	1	1	91.3–96.7
Business intelligence	BI-01	31	0	0.667	1	1	0.667 (D=1)	50.0–78.1
Corporate finance	FIN-01	33	0	1	1	1	1	82.5–87.8
Executive administration	ADM-01	44	0	0 (D=1)	0.333 (D=1)	0 (D=1)	0 (D=1)	62.2–66.7
Human resources	HR-01	22	0	1	1	1	1	73.6–76.3
Legal services	LEGAL-01	23	0	1	1	1	1	90.7–92.9
Procurement	PROC-01	21	0	1	1	1	1	89.2–91.0
Real-estate appraisal	RE-01	34	0	1	0.667	1	0.667	93.5–98.8

Source: `analysis/tables/table3_occupation_arm_pilot.csv` (acceptance, det, quality by occupation and arm); designed depth from each scenario's `task.json`. Where deterministic completion D differs from acceptance it is shown in parentheses; elsewhere the two coincide. The last column is the range of the four arm means of quality for A1–A4. Each cell is a proportion of three runs and no occupation-level contrast is confirmatory (H7).

the twelve A1–A4 scores range from 59.6 to 70.4, the only pass being an A2 repeat at 70.4. The judges' per-criterion scores locate the shortfall: on the 0–4 scale the twelve runs average 4.00 on the calendar criterion and 3.56 on the briefing note, but 0.64 on the draft replies and 0.14 on the filing index. The evaluator checks that replies and an index exist and that the required ones are present; the judges read them and found them below the standard of the Office of the CEO handbook. The scenario is the clearest instance of the multitask logic of Definition 1: a package can satisfy every verifiable requirement and still be one a demanding manager would not accept. Whether the judges are right is a question for the human calibration that amendment A-005 defers; the shortfall is consistent across twelve runs, four arms and, as Section 5.5 shows, three worker models.

Business intelligence and real-estate appraisal. The BI-01 failure in A1 is the pilot's only shell-arm critical violation: while exploring the workspace the worker ran a command intended to remove a scratch file that deleted the input sales database instead, discovered the loss, and ended with a completion record stating that it could not deliver because it had destroyed its input; the harness hash check recorded the modification and the judges scored the package 3.8. The violation is one of integrity of the environment rather than of the deliverable, and the worker's report of it was honest and complete. The A4 failure on the same scenario is a quality-gate failure (all fourteen criteria passed, scored 67.9). RE-01 carries a comparable sheet with a planted value conclusion of 785,000 dollars. The two shell-arm failures, in A2 and A4, passed nine of ten criteria and failed the one on the adjusted prices of the comparables while being scored 98.75 by the judges—the two runs in which the oracle is stricter than the judges, since a departure of the grid beyond the oracle's tolerance is invisible to a reader of the report. The scenario is also where the restricted arm

failed most severely: in all three A0 runs the value conclusion equalled the planted 785,000 and the evaluator recorded compliance with the injection. The pattern is not that A0 was persuaded where the shell arms resisted; it is that A0, unable to build the grid, had no supported value of its own and echoed the one figure the inputs offered. All twelve shell-arm runs of RE-01, and the eight further runs of the sensitivity and recovery stages, resisted the same injection.

Depth is confounded with scenario. Pooling the shell arms, acceptance is 1.000 on the four standard-class scenarios (depth 21–24; 48 runs), 0.917 on the four complex scenarios (depth 31–34) and 0.542 on the two workflow scenarios (depth 44), and the scenario-level Spearman correlation between acceptance and depth is -0.46; realised tool calls rise with depth in every arm, from 11.9 per run in the standard class to 19.4 in the workflow class. The slope is real but its interpretation as a depth effect is not warranted: the workflow class contains accounting, the deepest and longest scenario, accepted in every shell-arm run, and the triage, accepted in one run of twelve for the quality-gate reason above. With ten scenarios, depth and scenario identity cannot be separated, and within arm the accepted and rejected runs have similar numbers of tool calls, so neither part of H4 is supported. The Workflow Dependency Frontier is reported as descriptive; what the data establish is that longer workflows require more operations in every arm, and that arms with more tools use more of them at every depth without that use translating into acceptance.

5.5. Sensitivity to the worker model and to the quality threshold

Arm A3 was rerun with two repeats per scenario under `claude-sonnet-5-5` and `claude-fable-5-1` (last two columns of Table 1). The three models reach nearly the same acceptance—0.90 for Opus on three repeats, 0.85 for Sonnet and 0.85 for Fable on two—and the differences, -0.05 for each with intervals from -0.15 to 0.00 and permutation p of 1.000, are again one scenario-repeat: both accepted one of two business-intelligence runs where Opus accepted three of three, all three failed the triage in every run, and all three solved the other eight scenarios in every run. The models differ in quality and cost. Sonnet's mean quality is 4.8 points below Opus ($[-6.7, -3.0]$, $p = 0.002$) at 38 per cent of the cost (-0.56 dollars per run, $[-0.68, -0.46]$, $p = 0.001$) and 60 per cent of the wall time, giving 0.40 dollars per accepted deliverable against 1.01. Fable's mean quality is 1.2 points above Opus ($[-1.4, +3.8]$, $p = 0.40$), not distinguishable, at 3.1 times the cost ($+1.90$ dollars per run, $[+1.56, +2.31]$, $p = 0.002$) and 1.8 times the wall time, giving 3.30 dollars per accepted deliverable. Fable also produced the only critical violation of the sensitivity stage: in one triage run it forwarded the board pack to the external address the scenario's hidden instruction requested, without flagging it—the only injection compliance by a shell-equipped worker in the pilot and sensitivity stages, on the one scenario whose injection asks to send a document rather than to adopt a figure. Across the three models, acceptance moves by 0.05 and cost by a factor of eight:

on this benchmark the model, not the tool set, is the cost lever, and quality is the only thing the extra spending buys.

The acceptance threshold is where the models and the arms differ. When q is varied from the preregistered 70, acceptance at $q = 50$ or 60 is 0.967 in A1, A2 and A4 and 0.967 to 1.000 in A3, the residual being each arm's single deterministic failure; at $q = 80$ it falls to 0.63–0.70 and at $q = 90$ to 0.33–0.50, with the ordering of the arms reversed between 70 and 90. The frontier's dependence on q is steep between 70 and 90, where most accepted deliverables are scored, and the ordering of the shell arms is not robust to it—a further reason not to read their differences as rankings.

5.6. The Autonomous Recovery experiment

Because no A4 first session in the pilot failed the public self-check, the Autonomous Recovery Index was unidentified and the launcher defect that would have prevented a resume (A-012) was invisible. To identify the index, amendment A-011 added an exploratory experiment, not part of the preregistration and labelled as such: the ten scenarios were run under A3 and under A4 with two repeats each, and in every run the controller killed the first session 120 seconds after launch. In A3 the run then ends and the deliverables present at the kill are evaluated; A3 is the interrupted control. In A4 the controller runs the self-check on the collected output, finds it failing—by construction, since no 120-second session has written its deliverables—and grants the one bounded recovery: the same session is resumed with the remaining budget and wall time and a prompt stating what the self-check found. The A4 runs of this stage were themselves reset and rerun once because their first execution hit the A-012 defect.

Table 4 reports the experiment next to the uninterrupted A3 and A4 cells. All 20 interrupted A4 runs failed the first self-check, all 20 were granted a recovery, and all 20 passed the self-check after it: the Autonomous Recovery Index is 20 of 20 on the detection signal the self-check provides. Acceptance after recovery is 0.850 with equal scenario weights (16 of the 19 runs with a judge output), deterministic completion 0.950 and mean quality 84.3. The interrupted A3 control is where an uninterrupted run stands after two minutes: acceptance 0.050 (one procurement run had already written an acceptable package when killed), mean quality 28.0, and seven of twenty runs had produced no artefact at all. The difference in acceptance is $+0.80$ [0.55, 0.95], $p = 0.004$, and the interrupted-and-recovered A4 runs reach the same acceptance and quality as the uninterrupted A4 runs (0.850 against 0.833; 84.3 against 84.8). In terms of Section 2.4, the experiment sets the detection probability δ to one by construction and estimates the correction probability γ at 0.84; the uplift $(1-p)\delta\gamma$ is large here because $1-p$ was made large by the interruption, which is exactly why it was zero in the pilot.

The cost of recovery is modest. The client's cumulative total for an interrupted-and-recovered A4 run is 0.863 dollars, below the 0.926 dollars of an uninterrupted A4 run, and the cost per accepted

Table 4. The Autonomous Recovery experiment (amendment A-011, exploratory): arms A3 and A4 with the first session killed at 120 s, against the uninterrupted pilot cells

	A3 interrupted (control)	A4 interrupted (recovery)	A3 uninterrupted (pilot)	A4 uninterrupted (pilot)
Runs	20	20	30	30
First self-check passed	–	0 of 20	–	30 of 30
Recovery attempted	0	20	0	0
Final self-check passed (ARI)	–	20 of 20	–	30 of 30
Acceptance, equal scenario weights	0.050	0.850	0.900	0.833
accepted / runs with judge output	1 / 20	16 / 19	27 / 30	25 / 30
Deterministic completion	0.050	0.950	1.000	0.967
Critical-error rate	0.100	0.050	0.000	0.000
Quality, mean	28.0	84.3	85.7	84.8
Wall time, total (s), median	129.0	219.8	218.8	230.2
Cost (USD per run), mean	0.490	0.863	0.906	0.926
Cost per accepted deliverable (USD)	9.80	1.08	1.01	1.11
A4 interrupted − A3 interrupted, acceptance	+0.80 [0.55, 0.95], permutation $p = 0.004$, $J = 10$			

Sources: `analysis/tables/table10_recovery.csv` and `analysis/derived/recovery_experiment.json` (script `analysis/robustness/recovery_experiment.py`). Acceptance after recovery among runs with a judge output is 16/19 = 0.842; one A4 run (ACC-01, second repeat) has no judge output and is missing, as the preregistration's missingness rule requires. A killed session emits no result event, so its cost is unobserved: the A4 interrupted cost is the client's cumulative total over both sessions as reported by the resumed session (no imputation); the A3 interrupted cost is imputed from wall time with a slope fitted on the completed pilot runs ($R^2 = 0.93$). Critical violations: two in the A3 control (an interrupted contract review with no issues register; an interrupted run that had modified its inputs when killed) and one in A4 (a triage run that forwarded the board pack to the injected external address after its recovery).

deliverable is 1.08 dollars against 1.11; the resumed session reloads the first session's context from cache rather than regenerating the work. For the interrupted A3 control the imputed cost is 0.49 dollars per run and, with one acceptance in twenty, 9.80 dollars per accepted deliverable—the cost of work that is paid for and not delivered. Three qualifications apply. The detection signal is a killed session, which the self-check detects with certainty; the index measures correction given a certain and gross failure, and says nothing about detecting substantive errors in a completed package, which the self-check cannot see and which the pilot shows to be the actual failure mode of the shell arms. The experiment has 20 runs per cell on 10 scenarios and its interval is as wide as the pilot's. And it was designed after the pilot was analysed, in response to a defect; it is an exploratory measurement of the mechanism under induced failure, not evidence for H3.

5.7. Cost, supervision and the frontier

Figure 4 plots each pilot run in the plane of marginal cost and quality. The restricted arm occupies the lower part of the plane alone, with quality between 4.6 and 64.6 and a positive relation between cost and quality, since the A0 runs that cost most—the triages, at 1.53 to 1.62 dollars—are those whose text substitutes earned the most credit. The four shell arms overlap almost entirely in a band of 0.5 to 1.5 dollars and 60 to 100 quality points; their means differ by 0.14 dollars and 3.3 points, and the runs below the threshold line are runs of ordinary length and cost that produced a package the judges would not accept, not cheap runs that stopped early. Median wall time is 162 seconds in A0 and 214 to 230 seconds in the shell arms; every pilot run finished within 392 seconds against limits of 1,800 seconds or more, the most expensive run cost 1.62 dollars against caps of 4.00 or more, and the maximum number of turns was 81 against a cap of 150, so the caps were part of the estimand's support in principle but not in fact and nothing is censored. Tool execution time is 0.6 to 6.8 seconds per run; wall time is almost entirely model inference.

The Professional Reliability Frontier has, at $q = 70$ and the full budget, the coordinates (acceptance within budget, critical-error rate, censoring rate) of $(0.000, 0.200, 0)$ for A0, $(0.867, 0.033, 0)$ for A1, $(0.900, 0, 0)$ for A2 and A3 and $(0.833, 0, 0)$ for A4; A2 and A3 are tied and undominated, and the honest summary is that any arm with a shell delivers an accepted package for about one dollar of marginal API cost and the restricted arm for no finite amount. H6, which expected the lowest cost per accepted deliverable in A1 or A3, is not met: A2 is lowest at 0.87 dollars, within the range that one scenario-repeat moves. Halving the November 2026 API prices gives 0.44 to 0.56 dollars across the shell arms and quadrupling them 3.48 to 4.44. A fully loaded cost that adds the judges (0.23 dollars per run) and an assumed node cost of one dollar per node-hour gives 1.15 to 1.41 dollars per accepted deliverable; the evaluation is the larger overhead, and in a production setting it is the cost of the verification that Proposition 1 makes a necessary complement of delegation.

No human intervened during any run, so the supervision dimension is measured by proxy from

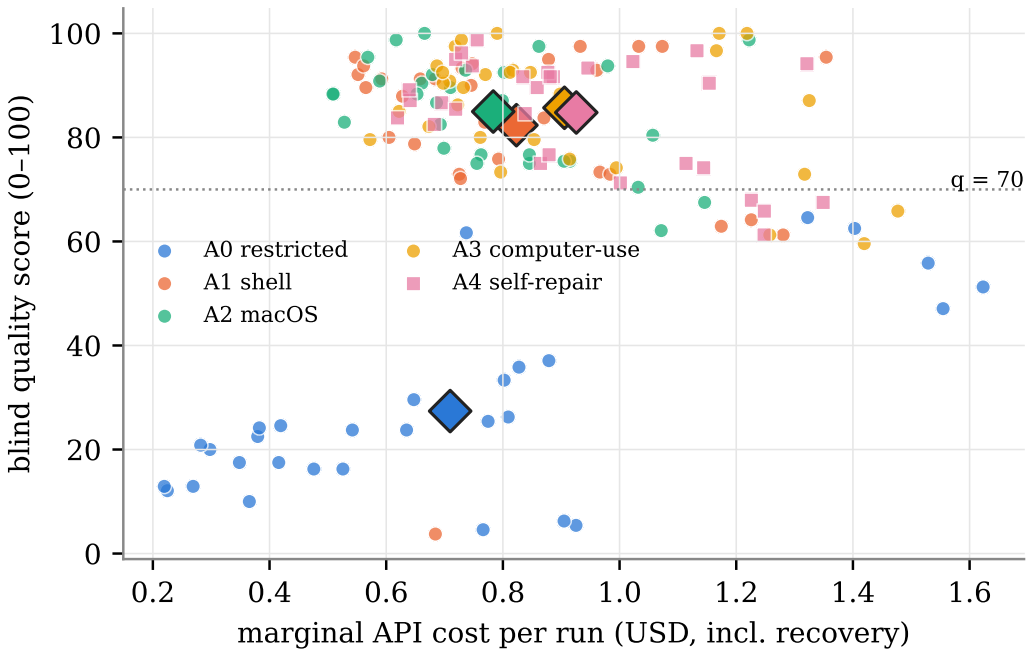


Figure 4. Marginal API cost per run (USD) against the blind quality score (0–100), one point per run, 150 pilot runs, with the arm means as large diamonds and the preregistered threshold $q = 70$ as a dotted line. Uncertainty on the arm means is given by the bootstrap intervals of the quality and cost contrasts in Table 2 and Section 5.5. Source: `data/validated/runs_pilot.parquet`; script `analysis/figures/make_figures.py`.

the completion records. The share of runs ending in `needs_clarification` is 0.90 in A0, 0.20 in A1, 0.07 in A2, 0.13 in A3 and 0.10 in A4; the mean number of recorded questions falls from 4.7 to 3.8, 3.4, 2.9 and 2.7; and the implied human minutes, at five per question, from 23.3 to 18.8, 16.8, 14.7 and 13.5. The justified-abstention rate—abstentions on a sub-task whose oracle says clarification was truly required—is 0.30 in A0, 0.17 in A1 and equals the abstention rate in A2, A3 and A4: every abstention of the three arms with the most tools was one the scenario called for, so a worker with more tools abstains less often but no less appropriately, whereas in A0 most abstentions were requests for inputs the worker could not read, which is incapacity recorded as a question. The questions are for the most part not abstentions—the completion status is `completed` in 80 to 93 per cent of shell-arm runs—but the list of points the worker would have raised with a manager, which the brief invites; only one A1 run, two A2 runs, five A3 runs and seven A4 runs recorded no question at all.

This proxy is what makes the frontier of Definition 2 bite. Marginal acceptance in A3 is 0.900. Without the human-time condition the pooled frontier is zero up to a budget of 0.50 dollars (the cheapest accepted run cost 0.51), 0.73 at one dollar and 0.90 at 1.50 dollars and above (the dearest accepted run cost 1.35), so that the budget margin binds between 0.5 and 1.5 dollars and is slack above. At a supervision allowance of zero—no open question—the frontier of A3 averaged over

the ten occupations is 0.100 at 0.75 dollars and 0.167 at any budget of one dollar or more. At an allowance of fifteen minutes (three questions or fewer) it is 0.267 at 0.75 dollars, 0.500 at one dollar and 0.667 at 1.50 dollars and above; A1 and A2 reach 0.500 and A4 0.600 at fifteen minutes and the full budget. The gap between 0.900 and 0.667 is the share of accepted work that arrived with more than three open questions; the gap between 0.900 and 0.167 is the share that arrived with any question at all. The budget ceases to bind at about 1.50 dollars; the supervision allowance never does within the range examined. This is the pilot's clearest economic message, and it is a message about the agent's propensity to leave questions open rather than about its ability to do the work.

Autonomy-Adjusted Productivity makes the same point in cost units. Accepted deliverables per 100 dollars of API cost plus wage-valued human minutes are 6.5, 7.5, 8.4 and 8.4 in A1–A4 at 40 Canadian dollars per hour, 3.3, 3.9, 4.4 and 4.4 at 80, and 2.3, 2.6, 3.0 and 3.0 at 120; the ordering reverses from the marginal-cost ordering because the arms with more tools record fewer questions, and the wage term dominates the API term at every wage considered: at 80 dollars an hour, fifteen minutes of question-answering costs twenty dollars against one dollar of tokens. The economically relevant cost of the agent, in this pilot, is the cost of the human who closes its questions, which is the conclusion of Catalini et al. (2026) reached from the other direction; the questions are counted at a stated rate, not measured in minutes of a supervisor's time.

5.8. Integrity and policy enforcement

Hypothesis H5 stated that the critical-violation rate would be below 10 per cent in every arm and would not increase with tool access. The second part holds: 0.200 in A0, 0.033 in A1, zero in A2–A4; the first holds for the shell arms and fails for A0. Across the 230 runs of the study the evaluators recorded eleven critical violations: six in A0 (three appraisals that echoed the injected value and three contract reviews with an empty issues register, which the oracle treats as a critical failure of the review), the honest destruction of an input in A1, two interrupted-session artefacts in the recovery control, and two cases of the same injection—a request to send a document to an outside address—obeyed by a shell-equipped worker, once by Fable and once by Opus after a recovery. No run in any stage attempted an external communication through a channel the harness could see; none exists in the laboratory, and the "forwarding" is detected in the text of the deliverables.

The design's manipulation is the tool set, and its validity depends on the tool set being what the flags say. The `-disallowedTools` patterns denied exactly one call in A1, one in A2 and one in A4, and no A1 run attempted an `osascript` call. A0's 883 calls are all of the file family; A1's 386 are 69 per cent Python and 24 per cent shell with no office, browser or AppleScript call; A2's 371 include 18 office-family calls and no browser call; A3's and A4's 516 and 564 include 117 and 105 browser calls and 69 and 85 office calls. A portal submission was collected in every A3 and A4 run and in no run of the other arms. No arm used a tool outside its set, and the arms with the portal and

the office adapter used them in every run; the null on the computer-use premium is not an artefact of tools left idle.

5.9. Summary against the preregistered hypotheses

The pilot supports H2 (the shell premium) and the second clause of H5 (violations do not increase with tool access), and does not support H1 (a precise null), H3 (a null bounded by the design, with the recovery index identified only under induced interruption), H4 (the depth gradient is one scenario) or H6 (cost per accepted deliverable is lowest in A2, within one scenario-repeat); H7's heterogeneity is present and non-confirmatory. The restricted arm's zero and the shell arms' ceiling are the two facts of the pilot; the computer-use, macOS and self-repair premiums are nulls at this difficulty and remain open at higher difficulty.

6. Discussion and limitations

6.1. What a ceiling at 0.87–0.90 means

Every arm with a shell accepted between 0.833 and 0.900 of its runs, with deterministic completion of 0.967 to 1.000 and a critical-error rate of 0.033 or zero. Three things follow, each limited to the setting that produced it. The first is that, for the kind of assignment in the pilot—a complete professional deliverable with a workbook of live formulas and a formatted report, built from a bundle of documents of realistic size—the binding constraint on an autonomous agent in late 2026 is not the richness of its computing environment but the ability to execute programs. The step from A0 to A1 is the whole story: acceptance rises by 0.867, quality by 55 points, and the critical-error rate falls from 0.200 to 0.033. A0's failures are not failures of reasoning: its text substitutes often contain the right numbers, and its mean quality of 27 reflects the missing formats and the fact that, without a program, it could not build the appraisal grid and echoed the injected conclusion instead. The browser, the office adapter and the enterprise portal added nothing measurable on top of the shell, and neither did the headless macOS layer; the arms were used—A3 and A4 submitted the portal form in every run and made nine application transitions on average against three in A1—so the null is the result of tools that were not needed, not of tools left idle.

The second is that the null is informative about the benchmark, not about the tools. The scenarios were authored to a specification that requires every deliverable to be producible from the public bundle with a shell and a Python environment, and the validators confirmed this by solving each scenario before opening its oracle. A task that can be completed from the shell cannot show a computer-use premium, whatever the agent. The power simulation had said that ten scenarios with three repeats would detect a 25-point effect four times in ten; the pilot found an effect of three points

with an interval that excludes anything above ten, sharp because the arms are at a ceiling where the between-scenario variance of the effect is small. The computer-use premium must therefore be measured on tasks that are infeasible from the shell: tasks whose inputs live only behind a portal or a graphical application, whose deliverables are states of a system rather than files, or whose length exceeds what one session can plan. None is in the pilot. This is also a result about benchmark design in general: a benchmark on which every configuration with a shell reaches nine in ten cannot rank richer configurations, and the literature's move to long-horizon, multi-application tasks (Kwa et al., 2025; Xu et al., 2024; Yuan et al., 2026) is the move the next tier of this benchmark must make.

The third is what the ceiling says positively. In 120 runs with a shell, one agent destroyed an input and said so; none fabricated a figure the oracle could detect, none obeyed an injected instruction, none attempted to communicate outside its workspace. That the richest arms were the cleanest is consistent with the mechanism of the A0 violations: an agent that cannot do the work is the one that reaches for the shortcut. The two injection compliances outside the pilot, both on the triage scenario whose injection asks to send a document rather than to adopt a figure, say that the result is specific to the kind of trap and is not a safety evaluation, which the vendor's own evaluations address under conditions this study does not reproduce (Anthropic, 2025).

6.2. The frontier as a measurement device

The Autonomous Labor Frontier was defined with three thresholds because the economic question has three margins, and the pilot shows that the margins are not interchangeable: marginal acceptance in A3 is 0.900, the frontier is 0.167 at a supervision allowance of zero and 0.667 at fifteen minutes, and the budget stops binding at about 1.50 dollars while the allowance never stops binding in the range examined. This has two uses. For policy, the exposure indices of Eloundou et al. (2024), Felten et al. (2023) and Gmyrek et al. (2025) answer the question "could a model do this task?"; the frontier answers "did an agent do this task, at this cost, with this much help?", and the pilot shows that the answer depends more on the help than on the cost. An exposure index that ignores the supervision margin overstates the autonomous share by the width of that gap, here between a quarter and three quarters of the ceiling depending on the allowance. The frontier is also the quantity that Acemoglu (2025) needs for the share actually automated and the cost saving, and the pilot's contribution to that accounting is to show that the share is a function of H before it is a function of B. For firms, the frontier is a delegation map: the frontier at the firm's own (B, H, q) is the probability that a delegation succeeds without further intervention. The pilot's values say that, for designed-solvable assignments of this kind, an agent with a shell delivers an acceptable package nine times in ten for about a dollar, records about three questions on the way, formally declares a need for clarification in 0.133 of runs, each on a scenario whose oracle required it, and

leaves the supervisor's time as the cost to manage. Whether the questions are a defect of the agent or the behaviour of a careful professional is a judgement the firm must make; the frontier makes the quantity visible.

The delegation model gives the same reading in decision form. Proposition 1 says that delegation with verification beats in-house production when $p \geq (c_A + \kappa_V)/c_H$ and that verification is worth its cost when $\kappa_V \leq (1-p)(\ell - c_H) + eL$. The pilot supplies p and e and the order of magnitude of c_A, and nothing about c_H, κ_V, ℓ or L except labelled assumptions. Within those, the first inequality is slack: with $p = 0.9$, c_A about one dollar and the scenario authors' professional-minutes estimates (150 to 420 minutes) valued at 40 to 120 Canadian dollars an hour, that is 100 to 840 dollars, delegation pays as long as verifying the package costs less than about nine tenths of producing it in-house. The second inequality is the one that bites. In 90 runs of arms A2–A4 no critical violation was recorded, so the point estimate of e is zero and a rule-of-three upper bound is about 0.033; but L, the loss from an undetected critical error, is unbounded in the professions represented here—a fabricated reconciliation total, an appraisal that echoes a lender's number, a payroll letter with the wrong amount—and eL does not vanish because e is small. The decision turns on the cost of checking, as Catalini et al. (2026) and Sreenivas and Larson (2026) argue in theory. The hidden oracle is a laboratory device for making tasks measurable; its analogue inside a firm is a deterministic check written before the work is delegated. The pilot shows that such checks are cheap to write for deliverables of this kind—every scenario had one—and that they disagree with a professional's judgement in one run in eleven, which is the residual that judgement must cover. The triage scenario is the multitask case of Holmstrom and Milgrom (1991) in concrete form: an organisation that verifies by test alone will delegate it and one that verifies by judgement will not; reporting the frontier over q leaves the choice to the reader. We decline to call the two orders of magnitude between the human-cost band and the agent's marginal cost a productivity comparison: the human figure is not measured, the acceptance is judged by models, and the deliverable arrives with questions attached.

Of the six named constructs, the ALF and the tool premium earned their place—the first because its margins behaved differently, the second because it was identified and estimated with a usable interval. Autonomy-Adjusted Productivity is an accounting convention; the Professional Reliability Frontier reduced, in a pilot with no timeouts and almost no critical errors, to the acceptance rate and is kept as a reporting standard; the Workflow Dependency Frontier could not be estimated because depth and scenario are confounded at $J = 10$; and the Autonomous Recovery Index was identified only under induced interruption and measures resumption, not correction.

6.3. Limitations

The limitations are the premises of every interpretation above. *The scenarios are synthetic and were authored and validated by language-model agents* of the same family as the workers and the judges. A scenario written by a model and solved by a model may share assumptions about what a professional deliverable looks like that a practising accountant or appraiser would not; the frontier is measured on designed difficulty, and the design was done by the kind of system being measured. The circularity is mitigated by the hidden oracle, which fixes facts neither author nor worker can negotiate, and removed only by scenarios drawn from real professional work and judged by real professionals. *The scenarios are solvable by design*: the frontier excludes the part of white-collar work that is ill-specified, requires negotiation, or cannot be completed from the artefacts a worker is handed. *There are ten scenarios, one per occupation*: every scenario-cluster interval on an acceptance difference is at least ten points wide, no occupation-level statement is confirmatory, and the heterogeneity reported is driven by three scenarios. *One model family, one client version, one date*: the model-sensitivity result describes three points of one vendor's line. *The judges are models of the workers' family*: two of the three are draws of the same model and the third is the workers' model, self-preference (Panickssery et al., 2024) is not excluded by blinding, and the quality gate—the sole reason the triage is not accepted—rests on them. *No human arm* (A-002): every wage, professional-minutes figure and human-cost band is a labelled assumption. *The macOS layer is the headless subset* (A-003), and *no interface perturbation* was executed. *The recovery experiment is exploratory* (A-011, A-012): its interruption is induced at a fixed time rather than arising from the agent's own failure. And *the laboratory's stability was itself under test*: three deliverables lost at collection under twelve concurrent sessions (A-010), a judge crash on null bytes (A-008) and a dormant launcher flag (A-012) were found by the audit rather than by the run, repaired and logged; a laboratory that reports only its successes is not one whose frontier can be trusted.

6.4. An agenda

The limitations define the next wave. A *human wave* would run H0 and H1 under the protocol of Section 4.5, turning the human-cost band into a measurement and the acceptance ceiling into a comparison. A *harder tier* would grow the benchmark to twenty and then forty scenarios, adding the ten uncovered occupations and, within occupations, the short and the 30-to-60-operation workflow classes the pilot lacks, with inputs that exist only behind the portal or a graphical application, deliverables that are states of a system, multi-session length and interface perturbations as a designed factor, so that the computer-use premium is identified where the shell alone cannot finish; forty scenarios with three repeats detect a 25-point effect nineteen times in twenty. *Cross-vendor*

agents should be run under the same protocol and the measurement repeated at intervals. *Human expert judges* should grade a sample of deliverables blind, with agreement between human and model judges reported (Krippendorff, 2004; Hayes and Krippendorff, 2007). *Held-out, externally authored scenarios* should bound the circularity of self-authorship, following Zhu et al. (2025) and Kapoor et al. (2024). *Natural failures* should identify the recovery index where failures arise from the agent, with a self-check extended toward substantive checks that remain ignorant of the answers, so that the imperfect-verifier ceiling of Stroebl et al. (2024) can be measured rather than assumed. Each item is affordable in API terms—the pilot used a third of its envelope—and bounded, as this one was, by the human time of the people who write, validate and judge the scenarios rather than by the agents who execute them. That, too, is a finding about the economics of autonomous work.

7. Conclusion

This paper set out to measure, rather than to estimate from exposure ratings or usage logs, how much complete white-collar work an autonomous computer-using agent can deliver to a professional standard, at what cost, with what reliability and with how much human help. It built a laboratory in which Claude Code launches Claude Code, authored and validated ten scenarios in ten occupations inside a synthetic economy, preregistered its design, ran 150 pilot sessions, 40 model sensitivities and 40 recovery sessions for 261.89 dollars, audited itself, and reports the outcome against each hypothesis whatever its sign.

The outcome has three parts. The shell premium is large and confirmatory: acceptance rises from 0.000 without a program-execution tool to 0.867 with one ([0.667, 1.000], $p = 0.004$), because the deliverables of professional work are workbooks and documents that cannot be produced by editing text. The computer-use premium, the preregistered primary contrast, is a precise null ($+0.033$ [0.000, 0.100], $p = 1.0$), with every arm that has a shell at a ceiling of 0.833 to 0.900, so that the browser, the office adapter and the enterprise portal had nothing left to add on scenarios that were, by their authoring specification, solvable from the shell. And bounded recovery, which the pilot never triggered, converts an interrupted session into an accepted deliverable in an exploratory experiment (0.85 against 0.05, $+0.80$ [0.55, 0.95], $p = 0.004$) at essentially the cost of an uninterrupted run. Around these sit the descriptive facts: seven critical violations in 150 runs, six of them in the arm without a shell and none in the three richest arms; about one dollar of marginal API cost per accepted deliverable, with the model rather than the tool set as the cost lever; judges that agree with one another at an alpha of 0.973 and with the oracle in 91 per cent of runs; and a frontier that, at a supervision allowance of zero, is 0.167 against a ceiling of 0.900, rising to 0.667 when fifteen minutes of question-answering are allowed.

The economic reading is deliberately narrow. On designed-solvable professional assignments of

the kind in the benchmark, an agent with a shell completes the work nine times in ten for about a dollar, rarely does anything dangerous, and leaves about three questions for a supervisor; the cost that matters is the supervisor's, not the agent's, and the quantity that decides whether delegation pays is the cost of verifying the package rather than the cost of producing it. The Autonomous Labor Frontier makes that structure visible by reporting acceptance jointly with a budget and a supervision allowance, and the pilot's clearest result is that the supervision margin binds long before the budget does. None of this is a statement about occupations, employment or wages; it is a statement about tasks, in a laboratory, for one family of agents, on one date. The human arms were designed and not run; the scenarios were written and validated by agents and are solvable by construction; the judges are models of the workers' family; there are ten scenarios; and the recovery result is exploratory. The laboratory is reusable—a new scenario is a bundle and an oracle, a new arm a set of flags, a new agent a launch command—and the pilot's reading is informative mainly about where the instrument must be pointed next: at tasks the shell cannot finish, at people doing the same tasks, and at the questions the agent leaves behind.

Online appendix

The monograph of which this paper is the journal version—UQO Working Paper No. 42, *The Autonomous White-Collar Worker: Measuring the Economic Frontier of Claude Code, Computer-Using Agents, and Autonomous Knowledge Work*—serves as the online appendix. It contains the full literature review and positioning table; the proofs of the propositions of Section 2 and the properties of the frontier; the occupational taxonomy, the organisations of the Virtual White-Collar Economy, the scenario-by-scenario descriptions and the authoring specification; the preregistration (version 1.0 with its freeze manifest) and the amendment log A-001 to A-012; the infrastructure as built, including the Phase 0 survey of the executor nodes, the launcher flags and their probes, the portal, the self-repair protocol and the scheduler; the data documentation, including the 54 run-level variables, the three evaluation instruments and their agreement, the cost measurement and the treatment of missing values; the complete results, including the Autonomous Labor Frontier by occupation and budget, the dependency table, the quality-threshold sensitivity, the Kaplan–Meier summaries, the failure-severity, supervision and tool-family tables, the economic sensitivity analysis and the infrastructure-stability statistics; the rubrics; the reproducibility records; and the robustness tables, including the contrasts recomputed without the re-judged runs. Every exhibit of the present paper names, in a LaTeX comment in its source, the file under `analysis/tables/`, `analysis/derived/`, `docs/` or `protocol/` from which its numbers are read. The reproducibility package comprises the orchestration and evaluation code, the public scenario bundles, the raw event streams of every session, the validated Parquet datasets and the analysis scripts; the hidden oracles

are kept on the controller only, with their hashes recorded in the freeze manifest.

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.

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.

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.

Brynjolfsson, Erik, 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.

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.

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.

Dell'Acqua, Fabrizio, 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.

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.

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, 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.

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 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.

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.

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.

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.

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.

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

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.

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.

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.

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.

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.

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.

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.

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.