# AgentAudit: How Much of an Agent Leaderboard Is Signal?

**Francisco Javier Gómez Pulido**

## Abstract

Agent leaderboards report one number per system, usually from one run per task. We ask how much of that number is signal, and how much of the usual statistical machinery survives when every task is run several times. We present AgentAudit, a typed and tested library (unbiased pass@$k$ and pass$^k$, task-level intervals, variance decomposition, paired tests with multiplicity control, ranking uncertainty, power planning, anytime-valid sequential comparison) and AgentAudit-Bench, an offline, deterministic, state-verified suite of 360 generated tool-use tasks. In simulations against known truth, pooling all runs gives 95% intervals that cover only 68.3% of the time with ten runs per task, and run-level tests that reject a true null in 17.9% of simulations; task-level procedures stay close to nominal, although the task bootstrap is optimistic with 20 tasks and extreme prevalence (down to 81%). Whether tasks or repeats are better at a fixed budget depends on the agent-by-task interaction: they are equivalent when it is small and tasks are better when it is large. Re-analysing 129 public SWE-bench Verified submissions, 28.3% of agent pairs are not distinguishable after Holm correction, and adjacent leaderboard positions are almost never separated (0/128 overall, 3/40 among entries that declare a single attempt); full coverage of the 500 instances could be verified for only 110 entries, and the share of separable pairs changes by less than two points under alternative treatments of missing logs. On tau-bench the gap between two frontier models is detected in retail for the mean-score estimand (not when runs are reduced to per-task majority outcomes) and not detected in airline, which is not evidence of equality. On 7 local models (10 runs on 120 tasks) run-to-run noise accounts for 34% of the variance of single-run outcomes, and a plug-in estimate says a 5-point difference would need about 387 tasks (a limited case study: only aggregates are stored and the run was not repeated).

## 1 Introduction

Public leaderboards for LLM agents, from SWE-bench (Jimenez et al., 2024) to $\tau$-bench (Yao et al., 2024) and the Holistic Agent Leaderboard (Kapoor et al., 2026), rank systems by a single success rate, often obtained from one run per task. Claims of a "new state of the art" are made on differences of one or two points. Three sources of randomness are rarely separated: *which tasks* were sampled, *which run* of a stochastic agent was recorded, and *how* the agent and the task interact. The principle that evaluations need error bars is established (Miller, 2024; Bowyer et al., 2025; Madaan et al., 2024; Bouthillier et al., 2021). What is less settled for agents is (i) which task-level procedures stay valid when every task is run several times, (ii) how a fixed budget should be split between more tasks and more repeats, and (iii) how much of an existing leaderboard survives a statistically careful reading.

**Terminology.** A *task* is one problem instance; a *run* is one attempt of one agent on one task (for SWE-bench entries we keep the leaderboard's own field name *attempts*, the number of runs per task a submission used to produce each answer); $R$ is the number of runs per task and $T$ the number of tasks. The *benchmark score* of an agent is the mean over tasks of the per-task success rate. *pass@k* is the probability that at least one of $k$ runs of a task succeeds, *pass*$^k$ the probability that all $k$ succeed; both are averaged over tasks.

**Research questions.**

**RQ1** When runs are clustered within tasks, which interval and test procedures keep their nominal coverage and error rate, and how badly do procedures that pool runs fail?

**RQ2** At a fixed budget of runs, is it better to spend it on more tasks or on more repeats, and what does the answer depend on?

**RQ3** How many differences between agents on public leaderboards (SWE-bench Verified, $\tau$-bench) are statistically distinguishable, and how sensitive is that count to what the public data do and do not document?

**RQ4** Does anytime-valid sequential comparison control false positives under continuous monitoring, and what does it cost in power?

**RQ5** On a controlled benchmark with several runs per task, how large is run noise relative to the score, and how reliable is the ranking of the evaluated agents?

Our working hypothesis was that many leaderboard differences are not distinguishable and that pass@$k$ hides unreliability that pass$^k$ reveals. We report where the evidence supports it and where it does not.

**Contributions.**

1. **AgentAudit**, a typed Python library with a CLI and a browser playground. Estimators are implemented from scratch and cross-checked against SciPy and statsmodels in the test suite (Section 3).
2. **AgentAudit-Bench**: six offline deterministic environments and 360 generated tasks with horizon and distractor knobs, verified by final state, with a seeded tool-failure injector and record/replay caching (Section 4).
3. **Pre-specified simulation studies** against exact ground truth for RQ1, RQ2 and RQ4, including robustness checks (prevalence, task heterogeneity, skewed paired differences, margin choice, near-tied leaderboards) that were written down before they were run (`docs/experiments.md`).
4. **A sensitivity analysis of public re-analyses**: SWE-bench Verified with its evaluation coverage and attempts audited, and $\tau$-bench with task-level inference (RQ3).
5. **A local-model case study** and the closed-form allocation rule of Section 3.6 (RQ5).

**Findings in brief.** (1) Pooling runs is not merely loose: with ten runs per task, a nominal 95% pooled Wilson interval covers 68.3% (68%–76% when tasks are very heterogeneous), while task-level intervals stay within 91.6%–95.5% (task bootstrap) and 93.2%–95.9% (beta-binomial) for $T \geq 50$; with $T = 20$ and one run per task at extreme prevalence the task bootstrap falls to 81%–82%. (2) Run-level tests reject a true null in 17.9% of simulations when the interaction is large; the task-level sign-flip test stays at or below nominal, and the bootstrap and $t$ tests are within about two points of it even for skewed differences. (3) At a fixed budget, tasks beat repeats only when agent-by-task interaction is non-negligible: the standard-error ratio of ten runs versus one is 0.96–1.04 for small and 1.54–1.56 for large interaction. (4) On SWE-bench Verified, 28.3% of pairs are not separable; adjacent positions are almost never separable, but the exact count depends on which entries are kept (Table 1). (5) A confidence sequence controls false positives under peeking (1.3% against 36.5% for naive peeking) at a price in power. (6) On local models, run noise is 34% of variance, the best model's pass$^{10}$ is 40.8% against a one-run score of 63.7%. 2 of the 7 models are near the floor (below 10%), where differences are trivially separable but uninformative about agent quality.

## 2 Related Work

**Statistics for LLM evaluation.** Miller (2024) gives standard errors, clustered errors, paired differences and power formulae for single-run question-answering evaluations; Bowyer et al. (2025) show that central-limit intervals fail for small benchmarks and recommend Bayesian or exact alternatives; Madaan et al. (2024)

quantify seed variance in training and evaluation; Biderman et al. (2024) document reproducibility pitfalls. Earlier work in NLP and RL made the case for significance testing, power and robust aggregation (Dietterich, 1998; Dror et al., 2018; Card et al., 2020; Agarwal et al., 2021; Bouthillier et al., 2021). The clustering argument itself is classical survey methodology (Kish, 1965). We build on this literature but target agents: several runs per task, outcomes verified by state, and the question of how to spend a budget.

**Agent evaluation and reliability.** $\tau$-bench introduced pass$^k$ to measure consistency across trials (Yao et al., 2024); Kapoor et al. (2024) argued that agent benchmarks must account for cost and reproducibility; HAL standardised large-scale agent evaluation (Kapoor et al., 2026). Huang (2026) propose an estimand-aware pairwise procedure for agent leaderboards and report unresolved close rank differences on SWE-bench and $\tau^2$-bench; Neuhof & Benjamini (2026) build rank intervals for MMLU from pairwise tests. Our contributions are complementary: a crossed variance decomposition over agents, tasks and runs, a closed-form allocation rule, validation against synthetic ground truth with robustness checks, an offline benchmark with controlled stochasticity, and an all-pairs re-analysis whose sensitivity to undocumented evaluation details is reported.

**Paired tests and multiplicity.** Sign-flip (randomisation) tests are exact under a symmetry assumption, which is stronger than a zero mean; their behaviour for the weaker null is studied by Romano (1990) and the studentised version by Janssen (1997). We use Holm adjustment (Holm, 1979), which controls the family-wise error under arbitrary dependence, and Benjamini–Hochberg (Benjamini & Hochberg, 1995), whose guarantee needs independence or positive dependence (Benjamini & Yekutieli, 2001); pairwise tests that share agents are not independent.

**Anytime-valid inference.** Confidence sequences (Howard et al., 2021) and betting-based constructions for bounded means (Waudby-Smith & Ramdas, 2024; Ramdas et al., 2023) permit continuous monitoring. Recent work applies them to LLM evaluation and online ranking (Hsu & Shekhar, 2026; Gu et al., 2026). The confidence sequence we use is the predictable plug-in, hedged betting construction of Waudby-Smith & Ramdas (2024) applied unchanged to paired per-task differences, so the construction is not a contribution of this paper; what we add is the explicit equivalence stopping rule and its error statement, an implementation audit (nonnegativity, predictability, grid), and a simulation of what it costs and buys relative to fixed-sample tests and to naive peeking. Hsu & Shekhar (2026) (sequential evaluation of a new model on a fixed question set, choosing which questions to query) and Gu et al. (2026) (online ranking from pairwise comparisons) use related e-process machinery for different designs; we do not compare against them empirically. Equivalence claims require a pre-declared margin (Lakens, 2017).

## 3 Preliminaries and Methodology

**Setting and estimand.** Agents $m = 1, \dots, M$ are evaluated on tasks $t = 1, \dots, T$ with $R$ runs each, giving scores $Y_{mtr} \in [0, 1]$ (binary in our benchmark). We regard tasks as an i.i.d. sample from a task population and runs as conditionally independent given agent and task. The estimand of an agent is the population benchmark score $\mu_m = E_t[p_{mt}]$, where $p_{mt} = E[Y_{mtr} \mid m, t]$. Agents are treated as *fixed*: the $M$ evaluated systems are not a sample from a population of agents, so the agent component below describes the spread of these $M$ systems rather than a population variance. Section 8 states what this does and does not license.

### 3.1 pass@$k$ and pass$^k$

With $c$ successes in $n$ runs of one task, $\binom{c}{k}/\binom{n}{k}$ is an unbiased estimator of $p^k$ (all $k$ runs succeed) and $1 - \binom{n-c}{k}/\binom{n}{k}$ is unbiased for $1 - (1-p)^k$ for $k \le n$ (Chen et al., 2021); unbiasedness is verified by exact summation over the binomial law in the test suite. We report benchmark-level means of these per-task estimators; the plug-in $\hat{p}^k$ is biased upwards. $k > n$ is refused rather than extrapolated. pass$^k$ is the reliability metric of $\tau$-bench (Yao et al., 2024); it decreases in $k$ whereas pass@$k$ increases.

### 3.2 Intervals for a benchmark score

Let $p_t$ be the success probability of task $t$, with mean $p$ and variance $\tau^2$ across tasks. The variance of the benchmark mean is

$$\mathrm{Var}(\bar{Y}) = \frac{\tau^2}{T} + \frac{p(1-p) - \tau^2}{TR}, \tag{1}$$

while a binomial interval on the pooled $TR$ runs assumes $p(1-p)/(TR)$. Their ratio is the design effect (Kish, 1965)

$$\mathrm{DEFF} = 1 + (R-1)\rho, \qquad \rho = \tau^2/\{p(1-p)\}, \tag{2}$$

so a nominal 95% pooled interval has asymptotic coverage $2\Phi(1.96/\sqrt{\mathrm{DEFF}}) - 1$. We implement Wilson (Wilson, 1927) and Clopper–Pearson (Clopper & Pearson, 1934) intervals (valid for one task or i.i.d. runs), the *task bootstrap* (resample whole tasks; percentile and BCa (Efron, 1987; Efron & Tibshirani, 1993)), a *runs-only* bootstrap (ignores $\tau^2/T$), a *two-stage* bootstrap (tasks, then runs; counts run noise twice), and an overdispersed beta-binomial profile-likelihood interval for the mean of $p_t \sim \mathrm{Beta}(\mu\kappa, (1-\mu)\kappa)$. The profile interval is a likelihood-ratio interval whose chi-square calibration relies on the beta mixing model; we verified the likelihood against an independent implementation, its reduction to the binomial likelihood interval at one run per task and its boundary behaviour, but we have no proof of validity for arbitrary task heterogeneity. In one additional simulation ($T = 50$, $R = 10$, 500 replications; Appendix F) it covered 95.4% when task difficulty was a two-point mixture, against 94.6% under a beta law (Monte Carlo s.e. about 1 point), so the assumption was not visibly critical there, which is not evidence for other mixing laws. BCa uses the bias correction $\hat{z}_0 = \Phi^{-1}$(share of replicates below the estimate) and the jackknife acceleration over tasks; the two-stage bootstrap resamples tasks and then runs within each drawn task, so its variance is that of the task bootstrap plus the within-task variance divided by $TR$ (verified analytically in the test suite), which is the sense in which it counts run noise twice.

### 3.3 Variance decomposition

We fit the crossed random-effects model $Y_{mtr} = \mu + a_m + b_t + (ab)_{mt} + e_{mtr}$ with variances $\sigma^2_m, \sigma^2_t, \sigma^2_{mt}, \sigma^2_e$ by the ANOVA method of moments (expected mean squares in Appendix A), in the spirit of generalizability theory (Cronbach et al., 1972; Brennan, 2001). Negative estimates are truncated at zero and the raw values are kept. For binary scores this is a linear-probability decomposition: exact for variances of means, not a latent-scale model. With $R = 1$ the interaction and run error are confounded and the library flags it. Two derived quantities matter in practice: the variance of the difference between two agents,

$$\mathrm{Var}(\bar{Y}_a - \bar{Y}_b) = 2\Big(\frac{\sigma^2_{mt}}{T} + \frac{\sigma^2_e}{TR}\Big), \tag{3}$$

in which the task main effect cancels, and the generalizability coefficient $E\rho^2 = \sigma^2_m/(\sigma^2_m + \sigma^2_{mt}/T + \sigma^2_e/(TR))$.

### 3.4 Paired comparison and multiplicity

For agents $a, b$ let $d_t = \bar{Y}_{at} - \bar{Y}_{bt}$; the task is the unit of analysis. We provide (i) a sign-flip permutation test of $\sum_t d_t$, (ii) a studentised sign-flip test of the paired $t$ statistic, (iii) a cluster bootstrap for the mean difference, (iv) exact McNemar (McNemar, 1947), (v) Cohen's $d_z$ and the probability of superiority $P(d > 0) + \frac{1}{2}P(d = 0)$. The plain sign-flip test is exact (conditionally on $|d_t|$) only under the *strong* null that each $d_t$ is symmetric about zero, which holds if the two agents are exchangeable within each task; it is not exact for the weak null $E[d] = 0$ with skewed $d$ (Romano, 1990), for which the studentised version is asymptotically valid by analogy with the studentised permutation theory of Janssen (1997), who treats the two-sample problem; we did not prove the one-sample result and check it by simulation (E1). It is enumerated exactly up to 14 non-zero differences; beyond that it is Monte Carlo with $n_{\mathrm{perm}} = 20{,}000$ seeded draws and p-value $(1 + \#\{|S^*| \geq |S|\})/(n_{\mathrm{perm}} + 1)$. McNemar's test needs one binary outcome per task: for $R = 1$ it is exact and conditional on the discordant pairs; for $R > 1$ we use the per-task majority outcome (strictly more than half of the runs; a task with exactly half is scored as a failure, an arbitrary convention). This changes the estimand to the difference of majority-success rates and discards the run-level information; it is a sensitivity analysis, not a conservative version of the primary test (Appendix F shows cases where it is less and more

significant than the mean-score analysis, depending on how ties are treated). Pairwise p-values are adjusted by Holm (Holm, 1979) (family-wise error under any dependence) or Benjamini–Hochberg (Benjamini & Hochberg, 1995) (false discovery rate under independence or positive dependence (Benjamini & Yekutieli, 2001); pairwise tests sharing agents are dependent, so Holm is our default).

**What a non-significant difference means.** Failing to reject $H_0 : \mu_a = \mu_b$ means a difference was not *detected* at this sample size. It is not evidence of equality. Equivalence is claimed only when a confidence interval or sequence lies inside a pre-declared margin (Lakens, 2017); "not distinguishable" in this paper never means "equivalent".

### 3.5 Ranking uncertainty

We resample tasks (the same tasks for every agent), recompute scores and ranks (ties broken uniformly at random) and report rank distributions, $P(\text{best})$, the matrix of $P(\bar{Y}_i > \bar{Y}_j)$ and the fraction of pairs that cannot be told apart. A split-half Kendall $\tau$ measures whether a fresh task sample would reproduce the ordering. Wide rank intervals show ranking uncertainty; they do not show that agents are equivalent.

### 3.6 Power and the tasks-versus-repeats trade-off

Let $s_t = 2\sigma^2_{mt}$ and $s_e = 2\sigma^2_e$. The mean paired difference has $\text{Var} = (s_t + s_e/R)/T$. At a fixed number of runs $B = TR$ this equals $(Rs_t + s_e)/B$, which is increasing in $R$ whenever $s_t > 0$: if tasks are free, a fixed number of runs is best spent on more tasks, and the advantage grows with the interaction $\sigma^2_{mt}$ (it vanishes as $s_t \to 0$). With cost $c_{\text{task}}$ per new task and $c_{\text{run}}$ per run, minimising the variance at budget $T(c_{\text{task}} + Rc_{\text{run}})$ gives

$$R^\star = \sqrt{\frac{s_e\, c_{\text{task}}}{s_t\, c_{\text{run}}}}. \tag{4}$$

Which cost scenario applies matters: a budget counted in runs, in money (tasks may need environment setup) or in wall-clock time (runs may share a machine) corresponds to different $c_{\text{task}}/c_{\text{run}}$. Power uses the noncentral $t$ distribution with $T-1$ degrees of freedom and the planner reports the minimum detectable effect. The variance components are themselves estimated, so tasks-needed and detectable-effect figures computed from a pilot are plug-in values with substantial uncertainty and should be read as orders of magnitude. Repeats remain essential for reliability metrics and for estimating $\sigma^2_e$.

### 3.7 Anytime-valid sequential evaluation

**Construction.** For per-task quantities $X_t = (d_t + 1)/2 \in [0, 1]$ with mean $m$, define for each candidate mean $m$ the capital processes $K^{\pm}_t(m) = \prod_{i \le t}\{1 \pm \lambda^{\pm}_i(m)(X_i - m)\}$ and the hedged process $K_t(m) = \frac{1}{2}\{K^+_t(m) + K^-_t(m)\}$. The bets are the predictable plug-in bets of Waudby-Smith & Ramdas (2024), $\lambda_i = \sqrt{2\log(2/\alpha)/(\hat{\sigma}^2_{i-1}\, i\log(1+i))}$, truncated at $\lambda^+_i(m) = \min(\lambda_i, \frac{1}{2}/m)$ and $\lambda^-_i(m) = \min(\lambda_i, \frac{1}{2}/(1-m))$; $\hat{\sigma}^2_{i-1}$ uses $X_1, \ldots, X_{i-1}$ only. Because every factor is at least $1/2$, both processes are nonnegative; if the $X_i$ are i.i.d. with mean $m$ they are martingales with initial value one (verified by exact enumeration over three-point laws in the test suite), hence so is $K_t(m)$. By Ville's inequality $P(\sup_t K_t(m) \ge 1/\alpha) \le \alpha$. The set $C_t = \bigcap_{s \le t}\{m : K_s(m) < 1/\alpha\}$ therefore contains the true mean at all times simultaneously with probability at least $1-\alpha$, so it may be inspected after every task and the experiment stopped at any data-dependent time (Howard et al., 2021; Ramdas et al., 2023). We report the hull $[\ell_t, u_t]$ of the retained candidates on a grid of 2001 means, widened by one grid cell, with $\ell_t$ non-decreasing and $u_t$ non-increasing. The hull is a superset of $C_t$; the widening guards against the grid, but the guarantee for means lying between grid points is numerical (resolution $5 \times 10^{-4}$ in $X$), not a proof. We verified bound invariance under a 20-fold finer grid (test suite).

**Assumptions.** The guarantee is for the mean of the task distribution: tasks must be i.i.d. draws from it and the $X_t$ bounded in $[0, 1]$ (the per-task difference of $R$-run means lies in $[-1, 1]$ for any $R$). Evaluating a fixed finite pool in random order is sampling without replacement; the guarantee then refers to the population mean only if the pool is regarded as an i.i.d. sample, which is the same assumption as in the rest of the paper.

Nothing is assumed about the distribution of $X_t$ beyond boundedness, and the run noise within a task is absorbed in $X_t$.

**Decision rule and what it guarantees.** Let $\Delta$ be the true mean paired difference, $\varepsilon \geq 0$ the margin and $[\ell_t, u_t]$ the interval for $\Delta$ obtained by mapping $C_t$ back to the difference scale. From task $t \geq t_{\min}$ the procedure stops with *a better* if $\ell_t > \varepsilon$, with *b better* if $u_t < -\varepsilon$, and with *equivalent* if $-\varepsilon \leq \ell_t$ and $u_t \leq \varepsilon$ (only possible for $\varepsilon > 0$, and only if the *whole* sequence, not the point estimate, lies inside the band). It stays *undecided* after the last task otherwise. Each wrong decision requires $\Delta otin[\ell_t, u_t]$ at the stopping time: a wrong "*a* better" means $\Delta \leq \varepsilon < \ell_t$; a wrong "*b* better" means $\Delta \geq -\varepsilon > u_t$; a wrong "equivalent" means $|\Delta| > \varepsilon$ while $[\ell_t, u_t] \subseteq [-\varepsilon, \varepsilon]$. All three are contained in the single event that the confidence sequence misses $\Delta$ at some time, so the probability of *any* wrong decision, of any of the three kinds, is at most $\alpha$ (not $2\alpha$ as in two one-sided tests), under optional stopping. This is a theoretical bound under the assumptions above; it is neither a guarantee of power nor of a specific rate such as "at most $x$%" for a given $\Delta$: the actual frequency at a given $\Delta$ can be far below $\alpha$ and is reported in Section 6.1 with Monte Carlo intervals. When $\Delta = 0$ and $\varepsilon = 0$ the procedure can never declare equivalence and ends undecided with probability near one minus the false-positive rate; when $|\Delta| \leq \varepsilon$ the sequence reaches "equivalent" only once its width is below $2\varepsilon$ (sample size grows like $\varepsilon^{-2}$), so small margins are unreachable at benchmark sizes; when $\Delta$ lies just outside the band, "equivalent" remains possible with probability at most $\alpha$.

## 4 AgentAudit-Bench

**Environments.** Six simulated worlds, all offline and deterministic: a virtual *file system*; an in-memory *SQLite* database; a REST-style *CRM* (customers and tickets); room *bookings*; *web forms* with validation and option codes; and a *tabular data* world with a restricted calculator. Each exposes 4–7 tools; tool errors are returned as observations, so an agent can recover from them.

**Tasks.** A procedural generator creates 360 tasks (60 per family) from a seed. Each family has five templates (for example, "move every `.txt` file into `/archive`", "upgrade every customer in a city"), and two knobs sweep a factorial grid: *horizon* (1–4 required operations or, for forms, 1–3 forms) and *distractors* (0–2: confusable files, decoy tables, same-surname customers, extra forms, extra rows). Free-text values are quoted in instructions to avoid punctuation ambiguity, a flaw found when auditing early traces. A balanced one-third *core* subset (120 tasks, 20 per family) is used for the local-model evaluation.

**Verification.** Success is decided by the final world state or the submitted answer, never by the agent's text. Every task is audited automatically in the test suite: a reference *oracle* must solve it, a do-nothing agent must not, and generation and resulting state must be deterministic in the seed. These checks establish that tasks are solvable and non-trivial, not that the checker accepts every valid alternative solution: only the state the generator anticipates is verified.

**Controlled stochasticity.** The benchmark includes a fault injector that fails each tool call independently with probability $q$, drawn sequentially along the call sequence from a generator seeded by (task, run) and independent of the agent; a failed call leaves the world unchanged and returns a transient 503, and injections are logged. Because draws are indexed by call number, two agents face the same failure pattern only if they make calls in the same order, so the injector gives matched failure *probability* and seeds, not identical failure events. The injector is implemented and tested but was not enabled in the experiments reported here.

**Agents and reproducibility.** Agents implement a small interface. All local models use one *uniform text tool protocol*: tools are listed as JSON in the system prompt, a call is any JSON object with `name` and `arguments`, one call per turn is executed, and generation stops after it. Using one protocol removes chat-template differences from the comparison; it also sidesteps a server-side tool-call defect we observed once (Appendix B). A record/replay cache keyed by agent configuration, message history, tools and seed makes an experiment replayable without the model, provided the cache is available; the cache and the raw trajectories of the local-model run are *not* in the repository, so the local-model numbers can be recomputed from the

stored aggregates but not from raw runs. The runner executes runs in run-major order, so any prefix of an interrupted evaluation is balanced, and resumes from checkpoints.

## 5 Experimental Protocol

All simulation experiments were specified before they were run (`docs/experiments.md`: question, design, result that would support the claim, result that would contradict it); unfavourable outcomes are reported below. Every random stream is seeded, the scripts are in `scripts/`, and every number in this paper is produced from the stored aggregates in `results/` by `scripts/make_paper_assets.py`; whether the aggregates themselves can be recomputed from raw outcomes differs by experiment (Appendix G).

**Simulated agents.** Synthetic agents follow a logit-normal item-response model, $p_{mt} = \sigma(\theta_m - b_t + g_{mt})$ with $b_t \sim N(0, \sigma_b^2)$ and agent-by-task interaction $g_{mt} \sim N(0, \sigma_g^2)$, and $Y_{mtr} \sim \text{Bernoulli}(p_{mt})$ independently across runs. Population scores, pass$^k$ and differences are computed by Gauss–Hermite quadrature, so coverage, error rates and power are measured against exact truth. The model is a convenient generator, not a claim about real agents; the robustness experiments vary its parameters.

**Experiments.** **S0** (RQ1, RQ2, RQ4, ranking calibration): the original coverage, type-I error, fixed-budget design, sequential and ranking simulations, 2000 replications per coverage cell. **E1** (RQ1): type-I error of the paired tests when $d_t$ is skewed with mean zero (weak null), $T \in \{20, 50, 100, 300\}$, $R \in \{1, 5\}$, 1500 replications. **E2** (RQ2): tasks versus repeats over task heterogeneity $\times$ interaction, 1500 replications. **E3** (RQ1): coverage across prevalence, heterogeneity, $T$ and $R$, 1000 replications. **E5** (RQ4): sequential stopping with margins $\{0, 0.03, 0.06, 0.10\}$ and minimum task counts, 2000 replications. **E6** (RQ3): pairwise distinguishability and Holm family-wise error on a six-agent leaderboard with an exactly tied pair, 400 replications. (The fixed-budget comparison was not repeated on the local-model data; see Section 6.4. S0 designates the simulations of the first version of this study; four of its five components (coverage, type-I error, fixed-budget design and ranking calibration) were re-run unchanged in the final audit and every stored numeric value was reproduced bit for bit, wall-clock times excepted (command and log in `FINAL_REPRODUCIBILITY_REPORT.md`). The fifth, the sequential comparison (S4), was not part of that comparison: it was regenerated after the confidence-sequence grid correction and its values differ from the first version's; the current script reproduces the current stored file bit for bit in a separate re-run. An earlier re-run during the first revision is supported only by commit history, not by a log.) Monte Carlo standard errors are about 0.6 points for a rejection rate near 5% with 1500 replications and 0.7 points for a coverage near 95% with 1000.

**SWE-bench Verified (RQ3).** Per-instance resolved lists of every submission in the public `swe-bench/experiments` repository that declares at least 50 resolved instances, 129 entries on 500 instances, analysed with exact McNemar tests and Holm/Benjamini–Hochberg adjustment (one binary outcome per task and entry). The public files do not state evaluation coverage or the number of attempts, so we audit them: an entry's coverage is *verified* when the union of its public log folders and its `no_generation` instances is exactly the 500 instances (110 entries; 17 have no public logs and 2 are incomplete), and the `attempts` field of its metadata gives one attempt for 41 entries, several for 80 (best-of-$n$ or multi-sample systems) and is missing for 8. "Verified" therefore means that log folders exist, not that the logs were inspected. We distinguish a documented failure (`no_generation`: the system produced no patch), an absent evaluation log (`no_logs`: the outcome is unobserved, and the official score counts it as unresolved), and a verified complete coverage record (folders plus `no_generation` account for all 500 instances and `no_logs` is empty). The headline analysis keeps the official convention for all entries; sensitivity analyses keep coverage-verified entries, single-attempt entries, and both.

**$\tau$-bench (RQ3).** Historical trajectories of gpt-4o and Claude 3.5 Sonnet (new, 2024-10-22 release), the first four trials per task, released by the authors under the MIT license. Only derived aggregates are stored in this repository.

**Local models (RQ5).** We evaluate 7 open-weight models served by Ollama on a single laptop (Apple M5, 16 GB) on the 120-task core subset of AgentAudit-Bench, 10 independent runs per task (8,400 runs),

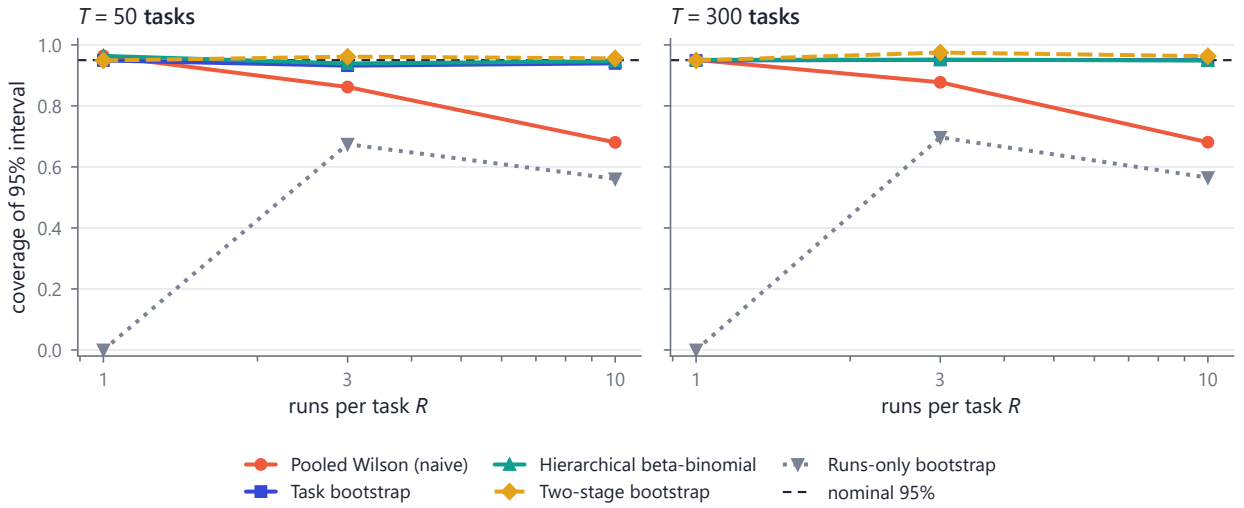



**Figure 1.** Coverage of nominal 95% intervals for the benchmark score as runs per task grow (task sd 1.5, interaction sd 0.5, true score $\approx 0.43$). Pooling all runs breaks down already at three runs.

temperature 0.7, with the text tool protocol of Appendix C so that every model is driven identically. The main run did not enable the fault injector (fault rate 0), so reliability under tool failure is not evaluated here.

## 6 Results

### 6.1 Simulation: validating the statistics (RQ1, RQ2, RQ4)

**Interval coverage (RQ1).** Figure 1 shows coverage of nominal 95% intervals for the benchmark score ($\sigma_b = 1.5$, interaction sd 0.5). With one run per task the pooled binomial intervals are fine (they are exact in that case). With three runs they cover 88.5% of the time and with ten runs only 68.3% (100 tasks; Monte Carlo s.e. 1.0 points), as the design effect predicts. A runs-only bootstrap is worse (56.4%). The task bootstrap, its BCa variant and the hierarchical beta-binomial interval stay close to nominal (task bootstrap: 94.8% at ten runs; BCa 94.8%; beta-binomial 95.1%), and the two-stage bootstrap is conservative (96.4%). Coverage of the task bootstrap for pass$^k$ ranges between 90.8% and 94.8%. Full numbers are in Appendix D.

**Robustness of coverage (E3).** The ranking of methods holds across prevalence and task heterogeneity, but the task bootstrap is not uniformly near nominal (Figure 2). For $T \geq 50$ and $R > 1$ it covers 91.6%–95.5% and the beta-binomial interval 93.2%–95.9%. With $T = 20$, one run per task and extreme prevalence with homogeneous tasks the percentile bootstrap covers only 81%–82% (BCa: from 80%) because the resampled score is nearly degenerate; the pooled Wilson interval is valid there and should be preferred. The runs-only bootstrap has zero coverage at $R = 1$ (0%) because it sees no variation. The pooled Wilson interval fails only when runs are clustered *and* tasks differ: at ten runs and task sd 1.5 it covers 68%–76%, and it is roughly nominal when tasks are nearly homogeneous. The grid is small ($T \in \{20, 50, 100\}$, $R \in \{1, 3, 10\}$, three prevalences, two levels of task heterogeneity, 1000 replications per cell, Monte Carlo s.e. up to 1.6 points), so we do not derive a sample-size threshold from it. What it supports is conditional: where runs are clustered within tasks the pooled interval should not be used, task-level intervals are the better default but are mildly optimistic even at $T \geq 50$ (coverage as low as 91.6%), and for very small single-run evaluations an exact binomial interval is preferable.

**Error rates of paired tests (RQ1).** With agent-by-task interaction sd = 1, $T = 100$ and ten runs per task, tests that treat each run as an independent pair reject a true null in 14.3% (two-proportion $z$-test) and 17.9% (run-level McNemar; Monte Carlo s.e. 0.9 points) of simulations, whereas the task-level sign-flip and bootstrap tests reject in 4.2% and 4.8% (Figure 3). With smaller interaction (sd 0.5) run-level McNemar is still liberal (8.7%). The pooled $z$-test is not always anti-conservative (it ignores pairing and is conservative at

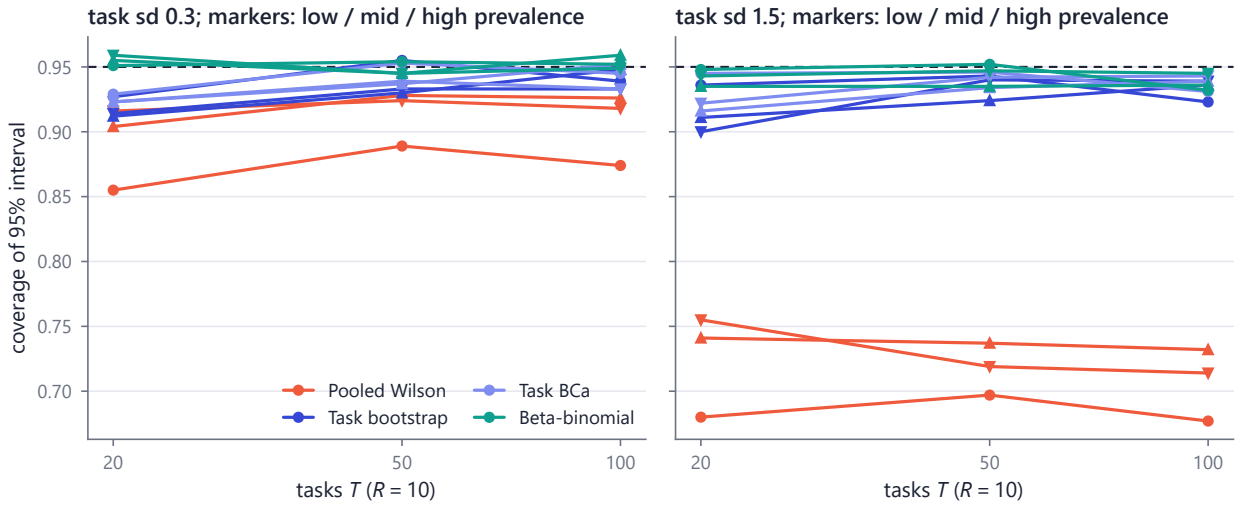



**Figure 2.** Coverage at ten runs per task for two task-heterogeneity levels and three prevalences (marker shape). Pooled Wilson loses coverage when tasks differ; the task-level methods approach the nominal level as $T$ grows.

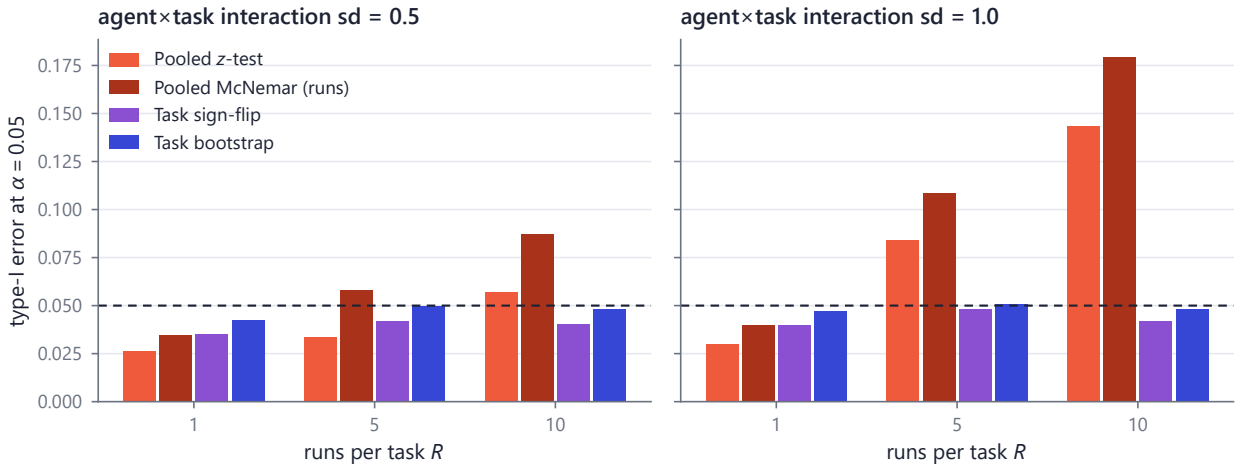



**Figure 3.** Type-I error at $\alpha = 0.05$ for $T = 100$ tasks and two agents of equal skill.

small $R$), so the harm of pooling is a loss of validity rather than a uniform inflation. Across all cells with a true null the sign-flip test rejects between 2.2% and 5.0%.

**Weak null with skewed differences (E1).** The simulation above has symmetric per-task differences, where the sign-flip test is exact. We also generated paired differences with mean zero but heavy skewness (a rare large win offset by frequent small losses; skewness up to 5.5). No procedure exceeded the pre-specified tolerance of 2 points over 5% at $T \geq 100$ (Figure 4): across all cells the largest rejection rates were 5.4% for the sign-flip test, 5.1% for its studentised version, 6.7% for the paired bootstrap and 6.7% for the paired $t$-test, with Monte Carlo error of about 0.6 points. The sign-flip tests are conservative on average (mean rejection 3.6%); the bootstrap and $t$ tests are on average at nominal (5.0%) but occasionally 1–2 points above it. We therefore keep the sign-flip test as the default, but we state its exactness only for symmetric differences.

**Tasks or repeats? (RQ2, E2).** In the original configuration (task sd 1.5, interaction sd 0.6) the standard error of the score difference at a fixed budget of 1200 runs grows by a factor of 1.60 between one and twenty runs per task, and power falls from 68.7% to 47.9% (ten runs) and 36.7% (twenty runs) (Figure 5); the analytic variance model predicts the simulated standard errors to within 3.8%. This is not a general law. Across nine configurations (E2, Figure 6) the standard-error ratio of ten versus one run per task is 0.96–1.04

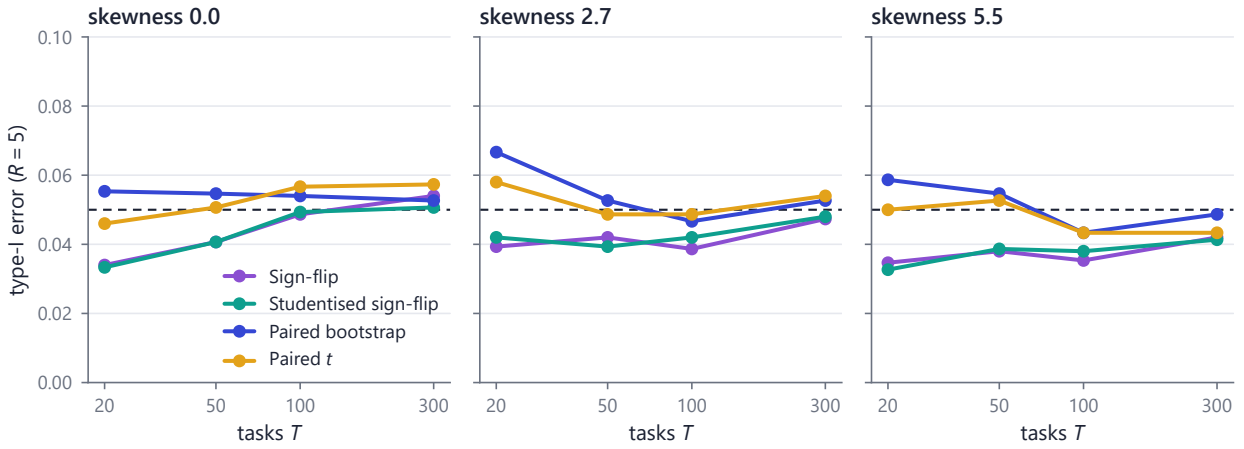



**Figure 4.** Type-I error at $\alpha = 0.05$ with $R = 5$ for paired differences with mean zero and increasing skewness (left to right).

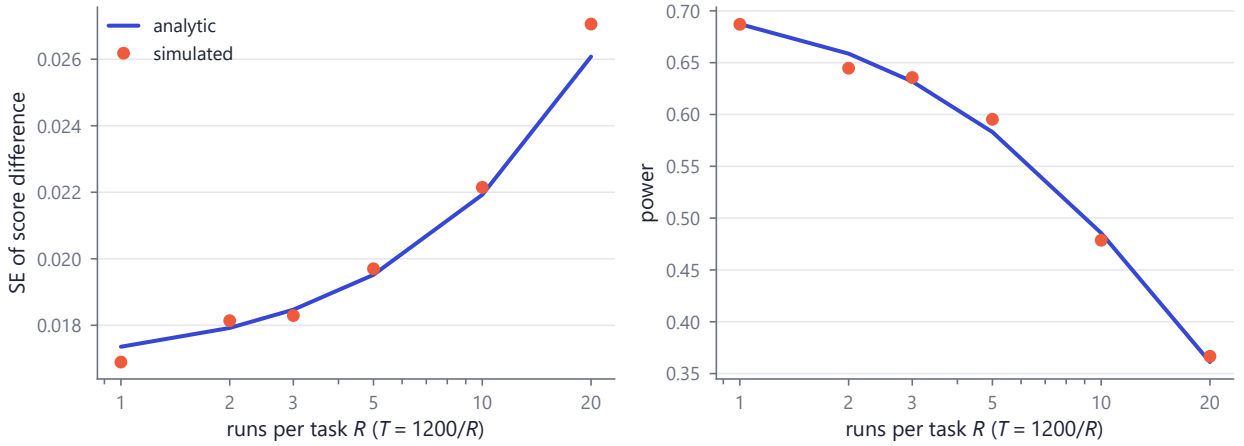



**Figure 5.** Fixed budget $B = 1200$ runs, $T = B/R$ tasks. Left: standard error of the difference between two agents; right: power of the paired $t$-test.

when the interaction sd is 0.2 (repeats cost nothing) and 1.54–1.56 when it is 1.0; the task standard deviation hardly matters, because the task main effect cancels in a paired difference. The cost-optimal number of runs per task for a task that costs ten runs is 32 with small interaction and 7 with large interaction. If the pool of tasks is capped (for example at 120), the question changes and the extra budget must go to repeats. Analytic and simulated standard errors agree to within 4.6% in E2. Repeats are not useless: they identify run noise and enable pass$^k$.

**Sequential evaluation (RQ4).** Applying a fixed-sample test after every task ("peeking") and stopping at the first $p < 0.05$ gives 36.5% false positives under the null, against 1.3% for the confidence sequence and 4.5% for the fixed-$n$ test (Figure 7; 600 replications per cell, Monte Carlo s.e. up to 2 points). The price is power: with a true difference of 7.5 points the sequence decides in 68.8% of runs against 94.5% for the fixed-$n$ test at the full budget of 300 tasks. When the effect is large it stops early (median 65 tasks for the largest difference). Confidence sequences buy the right to stop at any time; they do not make evaluation cheaper at a given power. Equivalence is only possible with a margin (E5, 2000 replications per cell). With margin zero "equivalent" was never declared (0.0%) and, for two agents of identical skill, the sequence stayed undecided in 98% of runs within 300 tasks: absence of a decision is not equivalence. With a margin of 10 points and a true gap of zero, equivalence was declared in 86% of runs (median 177 tasks). When the true gap exceeded the margin, equivalence was declared in at most 0.1% of runs per cell (95% Clopper–Pearson

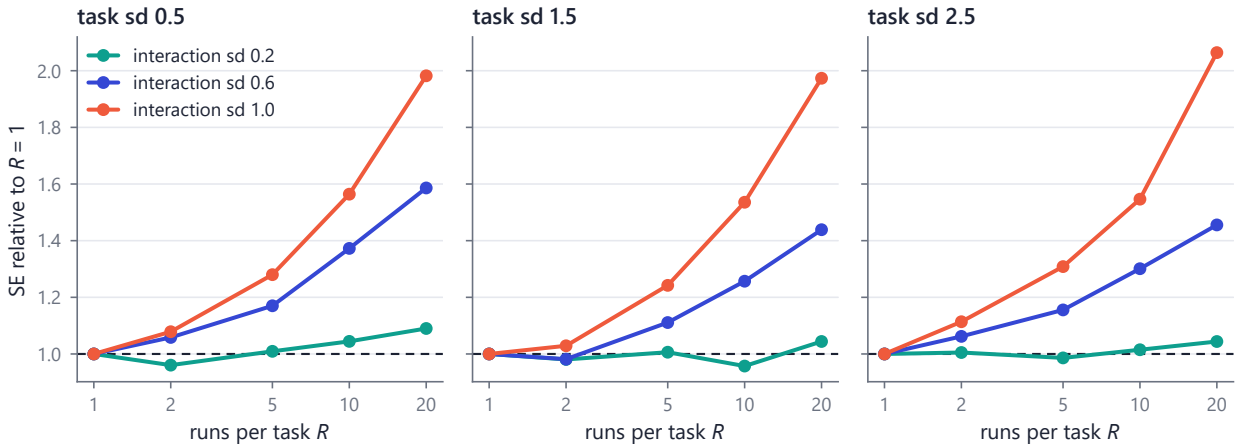



**Figure 6.** Standard error of the score difference at fixed budget relative to one run per task, for three task heterogeneities and three interaction levels. Curves are flat when the interaction is small.

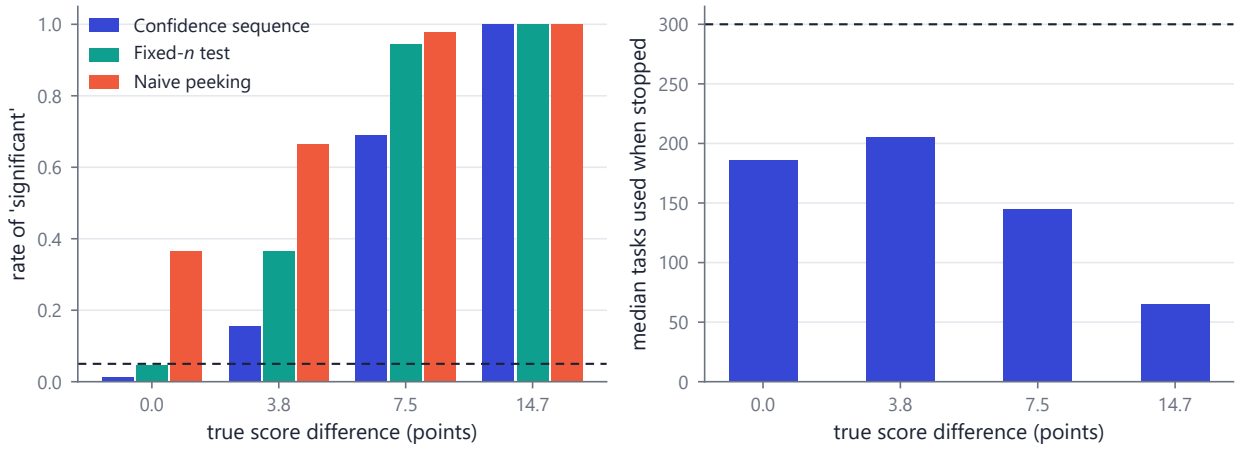



**Figure 7.** Sequential comparison, $R = 3$ runs per task. Left: fraction declared significant (the line marks 5%). Right: median number of tasks used when the sequence stops.

upper limit 0.44%, 2000 replications); this is a simulation frequency in the cells we ran, far below the bound $\alpha = 5\%$ that the theory gives for any bounded i.i.d. law (Section 3), and it is not itself a guarantee for other laws. A second simulation with a three-point law for the per-task difference (Appendix F, 2000 replications per row) found no wrong decision rate above 2.70% and a confidence-sequence miss rate of at most 2.90%, i.e. no violation of the 5% bound, so in these scenarios the sequence was conservative (the observed miss rate is well below $\alpha$; a low observed rate does not itself establish validity, which rests on the argument of Section 3). Reaching "equivalent" depends strongly on how dispersed the task differences are: with a margin of 10 points and a true gap of zero it was declared in 86% of runs in the main simulation but in only 9% of runs when the per-task differences are three-point and more dispersed. Narrow margins are rarely reachable with 300 tasks.

**Ranking uncertainty and near ties (RQ3, E6).** Bootstrap rank intervals are conservative for the population ranking (coverage 98.6%–99.9% for nominal 95%), and $P(\text{best})$ is reasonably calibrated, noisily at $T = 50$ (Figure 8). In a six-agent leaderboard with one exactly tied pair and one pair 2.7 points apart, Holm-adjusted tests rejected the tied pair in at most 2.0% of replications in every cell (the family-wise error is controlled and the procedure is conservative), but separated the close pair in only 19% of replications even with 500 tasks and three runs per task (and 0% with 50 tasks). Small gaps are, in practice, not resolved by realistic benchmark sizes.

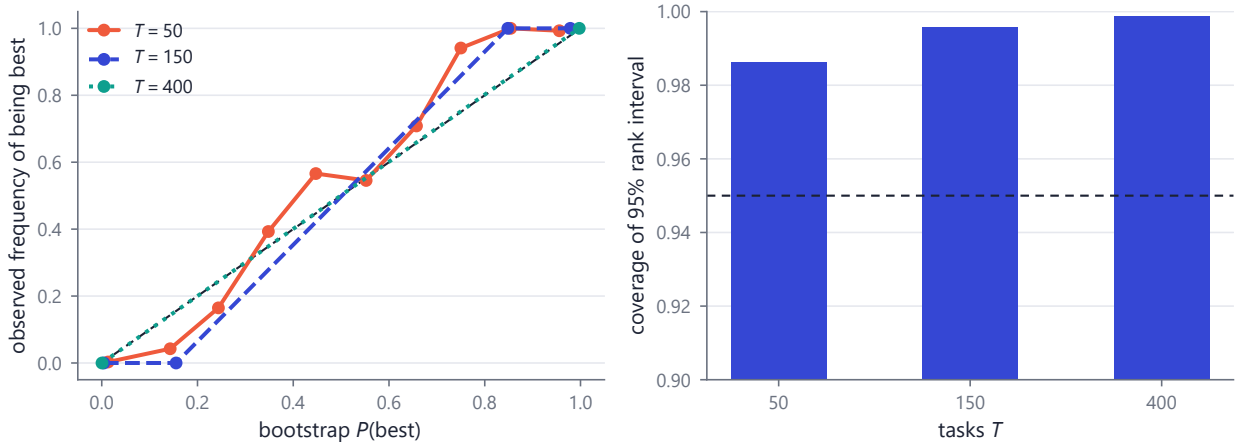



**Figure 8.** Six agents of known skill. Left: calibration of bootstrap $P(\text{best})$. Right: coverage of the 95% rank interval.

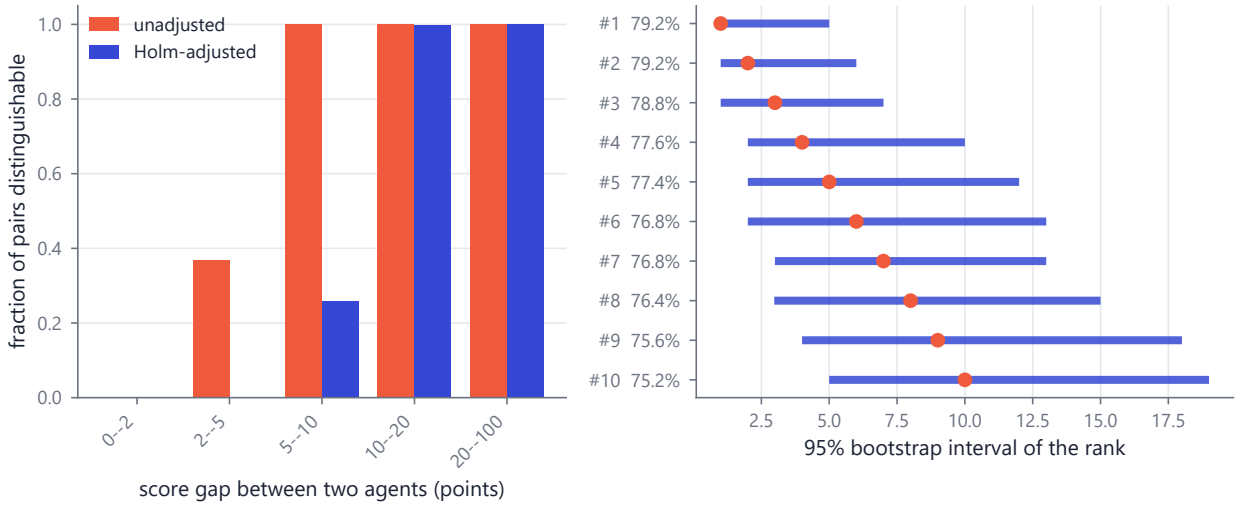



**Figure 9.** SWE-bench Verified, 129 public entries. Left: fraction of pairs distinguishable by score gap. Right: top-10 scores with 95% bootstrap rank intervals.

### 6.2 Re-analysis of public results (RQ3)

**SWE-bench Verified.** The 129 entries yield 8 256 agent pairs. After Holm adjustment 71.7% are distinguishable at $\alpha = 0.05$ (86.4% with Benjamini–Hochberg; 86.7% without correction), so 28.3% cannot be separated; "cannot be separated" means not detected, not equal. The result is driven by the score gap (Figure 9): pairs 2–5 points apart are separable in 0% of cases after correction, pairs 5–10 points apart in 26%, and larger gaps almost always. The smallest distinguishable gap is 6.8 points and the median is 23.2. The top entry (79.2%) has a 95% rank interval of 1–5 and probability 34% of being best.

**Sensitivity to undocumented evaluation details.** The headline count of adjacent positions depends on the entries retained (Table 1). Among all entries, 0/128 of adjacent leaderboard positions are distinguishable after Holm over *all* pairs (2/128 without correction). Restricting to entries whose coverage of the 500 instances could be verified gives 1/109; restricting to entries that declare a single attempt (41 entries) gives 3/40, presumably because fewer entries leave larger gaps between neighbours. The share of separable pairs is stable (71.7% coverage-verified, 69.3% single-attempt, 68.5% both). The qualitative conclusions, that most pairs but almost no adjacent positions are separable, therefore hold across the subsets, whereas the earlier claim that *none* of the adjacent positions is distinguishable is not robust: it is true only for the full set. The "coverage verified" label is weaker than it sounds: it means that public log folders plus the `no_generation` list account

**Table 1.** SWE-bench Verified sensitivity analysis. "Adjacent" counts distinguishable neighbouring leaderboard positions; Holm is applied over all pairs of the subset.

| Subset | Entries | Pairs separable (%) | Adjacent, Holm | Adjacent, unadjusted |
|---|---|---|---|---|
| All entries | 129 | 71.7 | 0/128 | 2/128 |
| Coverage verified | 110 | 71.7 | 1/109 | 3/109 |
| One attempt declared | 41 | 69.3 | 3/40 | 6/40 |
| Both | 37 | 68.5 | 2/36 | 5/36 |

for all 500 instances, and 36 such entries nonetheless list instances under `no_logs` (instances without an evaluation log, which the official convention also scores as unresolved). Overall 41 of the 129 entries have at least one such instance (up to 90); `no_generation` instances are genuine failures of the system, `no_logs` instances are unobserved. Appendix F therefore repeats the analysis on the 88 entries without missing logs (72.3% of pairs separable), with pairwise complete cases (71.4%) and with the least favourable filling of every missing log (70.8%): the share of separable pairs moves by less than two points (70.8% to 72.3%), so the headline is not an artefact of the missing-log convention. The inclusion rule (at least 50 resolved instances) removes only 6 of the 135 entries with a results file, so the results describe leaderboard rows that were publicly submitted and scored above that floor; they do not describe agents that were never submitted. The public repository changes over time; no snapshot hash was recorded when the files were downloaded (27 October 2026, from file timestamps). The latest commit touching `evaluation/verified` on the public repository, queried on 28 October 2026, is `92c5a4f` (1 September 2026), so the data are consistent with that state; this is an inference, not a verified identification. Many entries are best-of-$n$ or multi-sample systems and different scaffolds, budgets and dates are mixed; the comparison is of leaderboard rows, not of models.

**$\tau$-bench.** The estimand is the difference between the two models' benchmark scores (mean over tasks of the per-task mean reward over four trials); the unit of analysis is the task. We chose as primary analysis, from the estimand and design and written into the analysis script before the comparison table was computed, the studentised sign-flip test of the task-level differences, with the task bootstrap for the interval and Holm adjustment over the two domains. The plain sign-flip test is exact only if the task differences are symmetric about zero; the studentised version has no finite-sample exactness under the weak null $E[d] = 0$ and is supported by asymptotic arguments and by our skewness simulation (E1: $T \in \{20, 50, 100, 300\}$, $R \in \{1, 5\}$, skewness up to 5.5, rejection rate at most 5.1%). The observed task differences are only mildly skewed (skewness 0.11 retail, 0.29 airline), and every other procedure in Appendix F that targets the mean-score estimand gives the same conclusion, so the retail result does not hinge on this choice. For gpt-4o versus Claude 3.5 Sonnet the retail gap is +10.2 points (95% task-bootstrap interval [+3.9, +16.5]), primary $p = 0.002$ ($p = 0.005$ after Holm over the two domains); the plain sign-flip test gives $p = 0.002$ ($p = 0.005$ after Holm). The effect is detected in retail for the mean-score estimand, and every procedure in Appendix F that targets that estimand at the task level or with task clustering agrees. It is *not* detected when the data are reduced to per-task majority outcomes: exact McNemar gives $p = 0.16$ when a two-of-four task counts as a failure and $p = 0.096$ when the 22 tied tasks are dropped. These tests address a different estimand (a majority-success difference of 7.8 points instead of the mean-score difference), so the contrast shows that the conclusion depends on the representation of the data; it does not show that the majority test is conservative. McNemar tests on pass$^4$ and pass@4 give $p = 0.003$ and $p = 0.115$. The pooled-runs $z$-test ($p = 0.001$) is invalid here and is listed only for contrast. In airline the gap is +3.5 points [-6.0, +13.0] (a 20,000-draw bootstrap gives -6.0 to 13.5: bootstrap endpoints move by up to half a point between seeds) with primary $p = 0.545$ (pooled $p = 0.480$): not detected with 50 tasks, which does not show that the two models are equal. Plug-in power calculations suggest about 787 tasks would be needed at the observed airline gap (retail: 95), but each is one draw from a very wide distribution: a task bootstrap of the pilot gives 95% intervals of 59–11 215 (airline) and 38–591 (retail) tasks, so they are orders of magnitude, not predictions. Variance is dominated by run noise (58% airline, 46% retail) and task difficulty (32%, 37%), with agent-by-task interaction of 10% and 15%. Figure 10 contrasts pass@$k$ with pass$^k$: both models look strong when one success in $k$ suffices and much weaker when all $k$ must succeed. The data cover two models and two domains, and the model versions are those of the released trajectories.

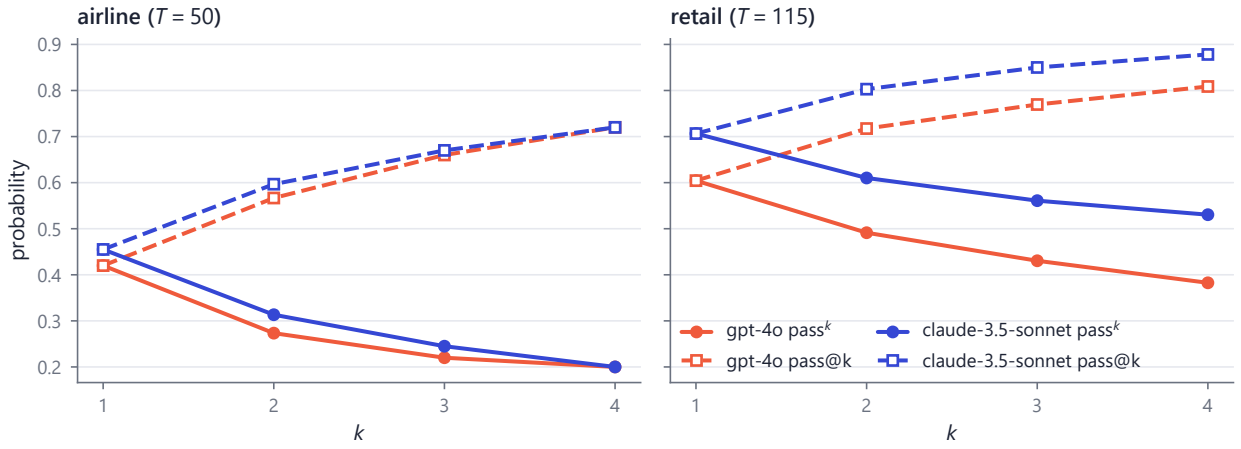



**Figure 10.** $\tau$-bench, four trials per task: pass$^k$ (solid) falls and pass@$k$ (dashed) rises with $k$.

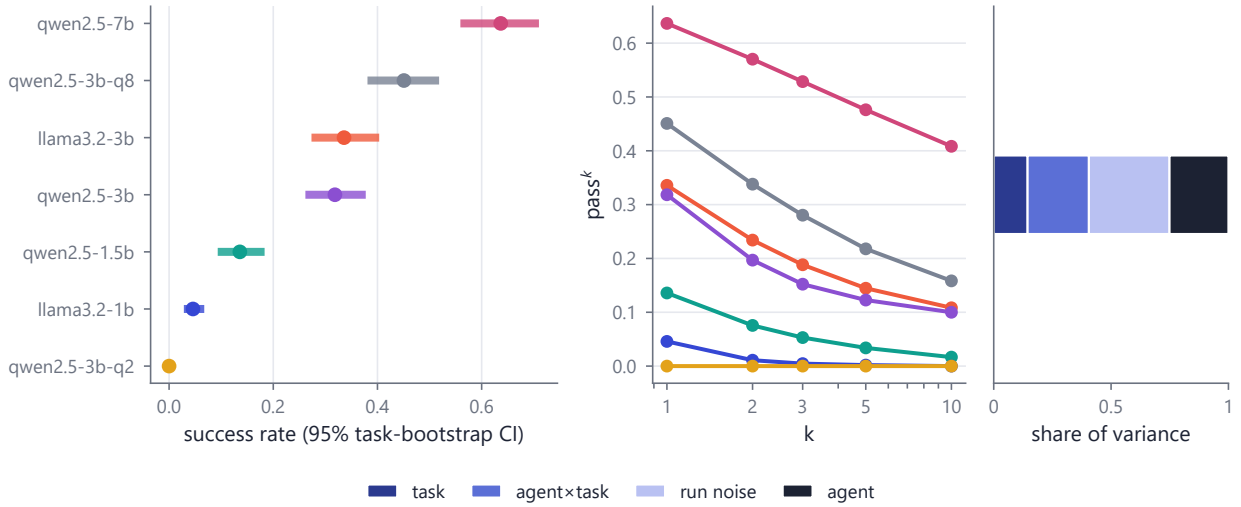



**Figure 11.** Local models on the 120-task core suite, 10 runs per task. Left: success rate with 95% task-bootstrap interval. Middle: pass$^k$. Right: variance decomposition of the full agents-by-tasks-by-runs array.

### 6.3 Local open-weight models (RQ5)

**Setup and headline numbers.** 7 models, 120 tasks, 10 runs each (8 400 runs in total). Success rates range from 0.0% (qwen2.5-3b-q2) to 63.7% (qwen2.5-7b); the full table with pass$^k$ and per-family scores is in Appendix E. Figure 11 (left) shows 95% task-bootstrap intervals. Scores here combine two things: protocol validity (does the model emit a parseable action in the uniform text protocol?) and task competence. The quantised variant `qwen2.5-3b-q2` never produced a parseable action (100% of its runs end with no tool call), so its score of zero measures protocol failure and says nothing about the underlying agent; we keep it as a degenerate control rather than dropping it silently, and it contributes to the variance decomposition and the pair counts as a floor. Without it, 1 of the 15 remaining pairs are not separable after Holm (recomputed from the stored pairwise $p$-values); the variance decomposition and the rankings without it could not be recomputed because the per-run outcomes are not stored (Appendix G). Backend errors (49 runs, 0.6%) are scored as failures. Results are specific to this protocol and these quantised models.

**Reliability is much lower than the headline score.** The best model's score of 63.7% falls to 52.9% for pass$^3$ and 40.8% for pass$^{10}$: a model that solves a task on average does not solve it every time. Run noise accounts for 34% of the variance of single runs, task difficulty for 14%, agent-by-task interaction for 26% and agent identity for 25%. The agent share describes the spread of these seven fixed models, not a

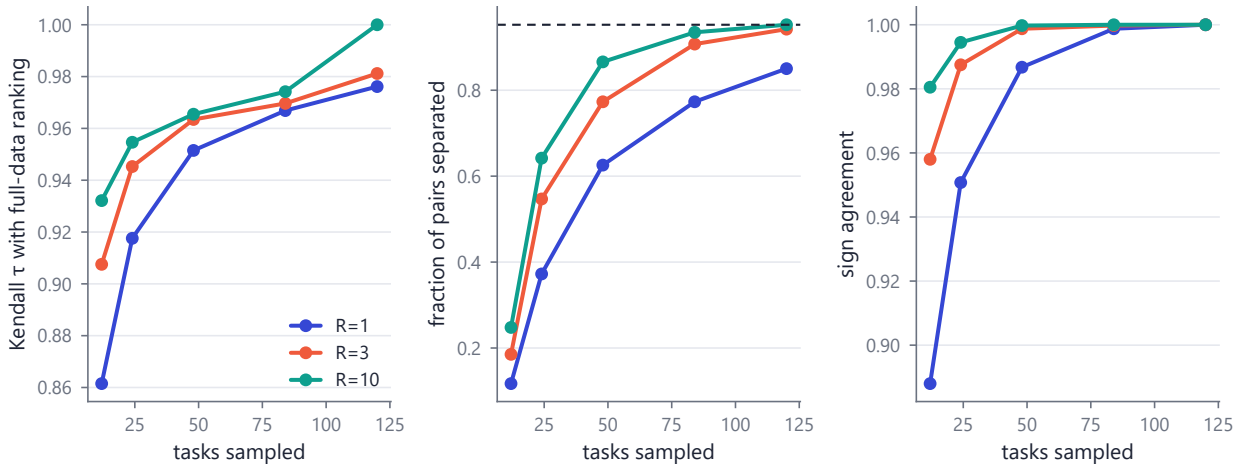



**Figure 12.** Sub-designs of the real data (random subsets of tasks and runs, 200 draws). Left: Kendall $\tau$ with the full-data ranking. Middle: fraction of the 21 pairs separated (paired $t$-test, Holm); the dashed line is the full data. Right: sign agreement for pairs that are significant in the full data. The reference is the full-data result, which shares observations with every subsample, so all panels measure stability relative to the observed sample, not accuracy against a known truth.

population variance. Run noise is of the same order as the quantity of interest, so a one-run evaluation is a noisy measurement even here.

**Ranking: the hypothesis is only partly borne out.** Because the models differ a lot in capability, the ranking is well determined: 1 of 21 pairs are not separable after Holm correction (llama3.2-3b versus qwen2.5-3b, a gap of 1.7 points, adjusted $p = 0.67$), and the split-half rank correlation is 0.94. This contrasts with SWE-bench, where scores are close; the statistical problems are most severe when systems are close, not absent otherwise. A plug-in power calculation says that resolving a 5-point difference between two of these agents at 80% power needs about 387 tasks at ten runs per task, more than three times the suite size, and the minimum detectable difference of the current design is 9.0 points; both are estimates from one pilot and carry the uncertainty documented for $\tau$-bench above. With the cost model of Section 3.6, the optimal number of runs per task is 4 when a task costs as much as ten runs.

### 6.4 Ablations on real data

**Tasks versus runs.** The sub-designs are random subsets of the same 120 tasks and ten runs, compared with the ranking and the significant pairs of the full data. The full-data ranking shares observations with every subsample and is itself an estimate, so this measures *internal stability relative to the full observed sample*, not recovery of an independently known population ranking; the pairs that are significant in the full data are also selected by the same data, so sign agreement among them is optimistic. Recovery of a known ranking is examined in the simulation (E6, Figure 8); a split-sample analysis on independent halves would require the per-run outcomes, which are not stored. Sub-sampling the real data (Figure 12), a ranking from all 120 tasks with one run per task has Kendall $\tau = 0.98$ with the full-data ranking and separates 85% of the pairs, against 95% with ten runs. With only 12 tasks and one run the ranking correlation is $\tau = 0.86$ and only 12% of pairs are separated. Since the large gaps are easy, rank correlation saturates quickly; the number of *separated pairs* is the informative measure and it keeps growing with both tasks and runs. We did not repeat the fixed-budget comparison of Section 6.1 on real data, so the simulation result on tasks versus runs is neither confirmed nor refuted here.

**Sequential stopping.** Applying the confidence sequence to tasks in random order, 93% of the 20 pairs that are significant on the full data are decided before the tasks run out, with a median of 29 tasks. This re-orders existing results; it is not a live stopping experiment. Large gaps are settled after a few tens of tasks; close pairs are not decided within the suite, as in the simulation.

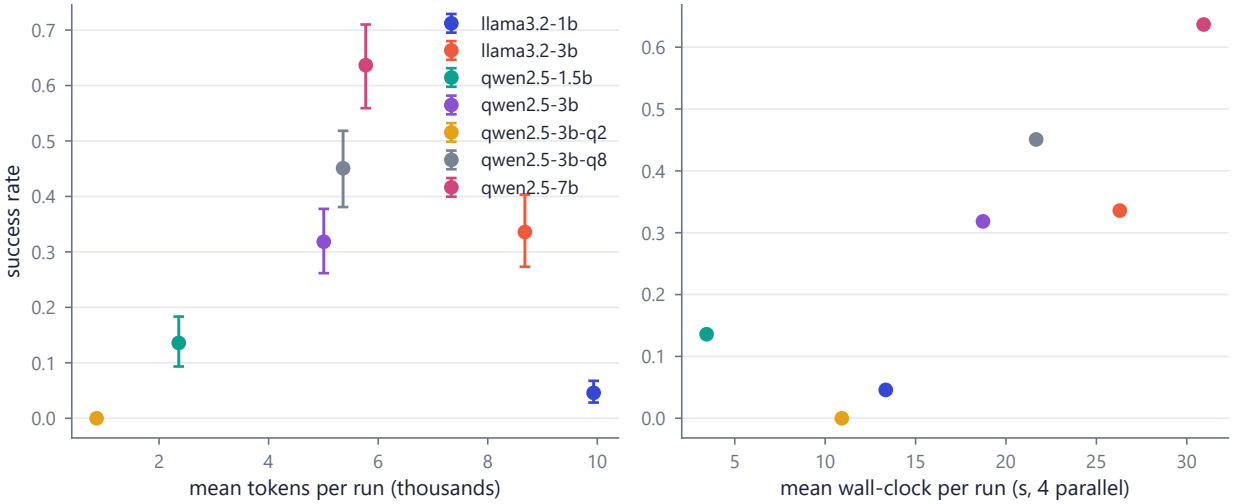



**Figure 13.** Success rate against mean tokens (left) and wall-clock (right) per run.

**Cost profile.** Figure 13 reports tokens and wall-clock per run. The weakest functioning model, llama3.2-1b, hits the step limit in 42% of runs and consumes the most tokens: failure is not cheap, so cost per success varies more than cost per run. Wall-clock was measured with four parallel workers on one machine and is indicative only.

## 7 Discussion

The clustering result is not surprising in isolation (clustered data need clustered inference), but its size is: reporting one pooled interval from several runs per task is common and, with ten runs per task, wrong by about twenty-five coverage points when tasks differ. The public re-analysis shows the consequence for leaderboards: with 500 instances and one run each, most pairs are separable but neighbouring entries almost never are, and small gaps are not resolved even with 500 tasks in simulation. None of this says leaderboards are useless: large gaps are reliably detected, and the top of the SWE-bench ranking is clearly above the bottom. It says that nearby rank positions should not be read as an ordering, and that "not distinguishable" is a statement about the evidence, not about the agents.

**Recommendations.** (i) Treat tasks as the sampling unit and report task-bootstrap or hierarchical intervals when $R > 1$ or $T \geq 50$ (in our simulation grid the task bootstrap is mildly optimistic, and the hierarchical interval relies on a beta model for task difficulty, Section 3.2); for very small single-run evaluations use an exact binomial interval. (ii) Compare agents on the same tasks with paired task-level tests and correct for multiplicity (Holm by default). (iii) Plan with the variance decomposition: prefer more tasks over more repeats when the agent-by-task interaction is non-negligible, prefer repeats when the task pool is capped or tasks are costly, and treat pilot-based task counts as orders of magnitude. (iv) Report pass$^k$ with pass@$k$. (v) If monitoring an evaluation as it runs, use a confidence sequence, not repeated fixed-sample tests, and pre-declare an equivalence margin if equivalence is the goal. (vi) Publish per-task, per-run outcomes and the number of attempts per submission, so that others can do this.

**Parts of the hypothesis that were weakened or not supported.** On local models the ranking is not fragile: with gaps of tens of points all but 1 of 21 pairs are separable and the split-half rank correlation is high, so the claim that rankings are unreliable is supported only for close systems (SWE-bench simulations and re-analysis) and not for widely spaced ones. The fixed-budget tasks-versus-runs result was not tested on real data, the real-data sequential analysis re-orders existing results rather than stopping a live run, and one model (q2) never produced a parseable action, so its score measures protocol failure rather than capability. Single hardware, single prompt protocol, small quantised models, no fault injection and raw trajectories not retained: conclusions do not transfer to frontier agents. Three further corrections to the first version of this

study: the claim that *no* adjacent SWE-bench position is distinguishable holds only for the full set of entries (Table 1); "more tasks beat more repeats" depends on the interaction and on whether the task pool is capped; and the sign-flip test is exact only for symmetric differences, although it did not over-reject for skewed ones.

## 8 Limitations, Ethics and Broader Impact

**Limitations.** *Inferential target.* Tasks are treated as i.i.d. draws from the population defined by our generator (or, for public data, from the benchmark), so intervals refer to that population and not to "ability"; tasks within a benchmark are often related (same repository, same domain), which would make our task-level intervals optimistic. Agents are fixed effects. *Local models.* The evaluation uses small quantised models, one machine (an Apple laptop reported by the author; hardware is not verifiable from the stored results), one prompt protocol, temperature 0.7 and no fault injection, so absolute scores and failure modes do not transfer to frontier agents. Raw trajectories and caches are not retained; the stored aggregates cannot be re-derived from raw runs. *Benchmark.* The environments are cleaner than real software tasks; checkers verify only the state the generator anticipates; difficulty knobs are not calibrated to any human notion of difficulty. *Statistics.* The variance decomposition is linear on the probability scale; the simulation model is one generator whose robustness we explored only along the axes listed in Section 5; confidence sequences require i.i.d. tasks and are conservative; plug-in power and tasks-needed figures are highly uncertain; Monte Carlo errors are 0.6–0.7 points. *Public data.* The SWE-bench files do not state coverage or attempts; we audit them but 17 entries have no public logs. Scaffolds, budgets and dates are mixed. The $\tau$-bench data cover two models and two domains.

**Ethics and impact.** No human subjects or personal data are involved; environments are synthetic and public data are used under their licenses ($\tau$-bench: MIT; the SWE-bench experiments repository declares no license, so we store only derived aggregates and a download script). Better error bars can reduce overclaiming, but rigour can also be used to dismiss real improvements; a non-significant result is not evidence of equivalence, which is why equivalence requires a margin. All experiments used free, local, open-weight models and public data; no paid API was used.

## 9 Conclusion

Agent leaderboards carry less information than their precision suggests, most of all for close systems. Treating the task as the sampling unit, testing the procedures against ground truth including where they fail, and auditing what public data do and do not document give a more defensible reading of agent results. We release the library, the benchmark, the scripts and the stored results so that every table, figure and number can be regenerated from the stored aggregates in `results/`; the simulations and the public-data re-analyses can be rerun from the scripts, whereas the local-model run cannot be repeated from the repository (Appendix G).

## References


Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. arXiv:2108.13264.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B*, 57(1):289–300, 1995.

Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 29(4):1165–1188, 2001.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, et al. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*, 2024.

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, et al. Accounting for variance in machine learning benchmarks. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 3, 2021. arXiv:2103.03098.

Sam Bowyer, Laurence Aitchison, and Desi R. Ivanova. Position: Don't use the CLT in LLM evals with fewer than a few hundred datapoints. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2503.01747.

Robert L. Brennan. *Generalizability Theory*. Springer, 2001.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. With little power comes great responsibility. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9263–9274, 2020.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

Lee J. Cronbach, Goldine C. Gleser, Harinder Nanda, and Nageswari Rajaratnam. *The Dependability of Behavioral Measurements: Theory of Generalizability for Scores and Profiles*. Wiley, 1972.

Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1383–1392, 2018.

Bradley Efron. Better bootstrap confidence intervals. *Journal of the American Statistical Association*, 82 (397):171–185, 1987.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

Runzhe Gu, Wenguang Sun, Bowen Gang, and Xintao Xia. Anytime-valid inference for online ranking of large language models. In *Proceedings of the International Conference on Machine Learning (ICML), PMLR 306*, 2026.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2): 65–70, 1979.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49(2):1055–1080, 2021.

Chia-Yu Hsu and Shubhanshu Shekhar. Efficient sequential evaluation of large language models. *arXiv preprint arXiv:2607.17409*, 2026.

Wei-Jung Huang. What does an LLM-agent leaderboard rank actually compare? *arXiv preprint arXiv:2609.07785*, 2026.

Arnold Janssen. Studentized permutation tests for non-i.i.d. hypotheses and the generalized Behrens–Fisher problem. *Statistics & Probability Letters*, 36(1):9–21, 1997.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.06770.

Sayash Kapoor, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. AI agents that matter. *arXiv preprint arXiv:2407.01502*, 2024.

Sayash Kapoor, Benedikt Stroebl, Peter Kirgis, Nitya Nadgir, Zachary S. Siegel, et al. Holistic agent leaderboard: The missing infrastructure for AI agent evaluation. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2510.11977.

Leslie Kish. *Survey Sampling*. Wiley, 1965.

Daniël Lakens. Equivalence tests: A practical primer for $t$ tests, correlations, and meta-analyses. *Social Psychological and Personality Science*, 8(4):355–362, 2017.

Lovish Madaan, Aaditya K. Singh, Rylan Schaeffer, Andrew Poulton, Sanmi Koyejo, Pontus Stenetorp, Sharan Narang, and Dieuwke Hupkes. Quantifying variance in evaluation benchmarks. *arXiv preprint arXiv:2406.10229*, 2024.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations. *arXiv preprint arXiv:2411.00640*, 2024.

Bitya Neuhof and Yuval Benjamini. Quantifying ranking uncertainty in LLM benchmarks. *arXiv preprint arXiv:2607.16259*, 2026.

Aaditya Ramdas, Peter Grünwald, Vladimir Vovk, and Glenn Shafer. Game-theoretic statistics and safe anytime-valid inference. *Statistical Science*, 38(4):576–601, 2023.

Joseph P. Romano. On the behavior of randomization tests without a group invariance assumption. *Journal of the American Statistical Association*, 85(411):686–692, 1990.

Ian Waudby-Smith and Aaditya Ramdas. Estimating means of bounded random variables by betting. *Journal of the Royal Statistical Society Series B*, 86(1), 2024. doi: 10.1093/jrsssb/qkad009.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. $\tau$-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

## A Mathematical details

For balanced data with $M$ agents, $T$ tasks and $R$ runs, the expected mean squares of the crossed model are

$$E[MS_m] = \sigma_e^2 + R\sigma_{mt}^2 + TR\sigma_m^2, \qquad E[MS_t] = \sigma_e^2 + R\sigma_{mt}^2 + MR\sigma_t^2,$$
$$E[MS_{mt}] = \sigma_e^2 + R\sigma_{mt}^2, \qquad E[MS_e] = \sigma_e^2,$$

and the estimators solve these for the components, truncating negative values at zero (the raw values are retained). With $R = 1$ the interaction and error are confounded; the library flags this. For one agent, the one-way model gives $\hat{\sigma}_t^2 = (MS_b - MS_w)/R$ and $\hat{\sigma}_e^2 = MS_w$, which equals the REML estimate for balanced data (verified against statsmodels in the test suite).

**Unbiasedness of pass$^k$.** If $C \sim \mathrm{Bin}(n, p)$ then $E\binom{C}{k} = \binom{n}{k}p^k$, so $\binom{C}{k}/\binom{n}{k}$ is unbiased for $p^k$; similarly $E\binom{n-C}{k} = \binom{n}{k}(1-p)^k$.

**Pooled design effect.** Writing $\bar{Y} = \frac{1}{T}\sum_t \hat{p}_t$ with $\hat{p}_t \mid p_t \sim \mathrm{Bin}(R, p_t)/R$, the law of total variance gives $\mathrm{Var}(\bar{Y}) = \tau^2/T + E[p_t(1-p_t)]/(TR)$ and $E[p_t(1-p_t)] = p(1-p) - \tau^2$.

**Optimal runs.** Minimising $(s_t + s_e/R)(c_{\text{task}} + Rc_{\text{run}})$ over $R > 0$ gives $R^\star = \sqrt{s_e c_{\text{task}}/(s_t c_{\text{run}})}$; the planner evaluates the two neighbouring integers.

## B Tool-protocol defect observed in Ollama

With Ollama 0.40.8 (macOS, Apple silicon), the native `/api/chat` tool-calling interface returned an empty message (no content, no tool calls) for `qwen2.5:3b` on a file-creation task when eight tools were declared, although the raw generation obtained from the `/api/generate` endpoint with the model's own template contained a well-formed `<tool_call>` block. The failure was deterministic across seeds and temperatures in the author's tests; it disappeared with a single declared tool or without a system message. The logs of these tests are not part of the repository and the observation was not re-run during the revision of this paper, so it is reported as an unverified observation about one server version, not about the model. The experiments use the uniform text protocol of Section 4 regardless.

## C Prompts

**System prompt.**

```
You are an autonomous assistant that completes tasks by calling the provided tools. Work
step by step, use the tools to inspect the current state before changing it, and make
exactly the requested changes and no others. If a tool returns an error that looks
temporary (for example a 503), retry the call. When the task is done, call finish; if
the task asks a question, call submit_answer with the answer.
```

**Tool section appended to the system prompt.**

```
# Tools

You can call these tools (JSON signatures):
<tools>
{"name": "TOOL_NAME", "description": "DESCRIPTION", "parameters": {"type": "object",
    "properties": {"ARG": {"type": "string", "description": "DESCRIPTION"}}, "required":
    ["ARG"]}}
</tools>

To call a tool, reply with exactly one block of this form and nothing else:
<tool_call>
{"name": "<tool name>", "arguments": {<arguments as JSON>}}
</tool_call>
You will receive the result inside <tool_response></tool_response>. Call one tool at a
    time.
```

## D Full coverage table

**Table 2.** Coverage (%) of nominal 95% intervals for the benchmark score; wilson, CP: pooled binomial intervals; the rest respect the task structure. $T$: tasks, $R$: runs per task.

| $T$ | $R$ | Wilson | CP | Task boot. | BCa | Two-stage | Beta-bin. |
|---|---|---|---|---|---|---|---|
| 20 | 1 | 94.9 | 94.9 | 91.2 | 94.0 | 91.2 | 94.9 |
| 20 | 3 | 87.9 | 89.6 | 93.3 | 94.7 | 96.5 | 95.0 |
| 20 | 10 | 68.2 | 70.2 | 92.5 | 94.2 | 95.2 | 93.8 |
| 50 | 1 | 96.4 | 96.4 | 94.9 | 96.2 | 94.9 | 96.4 |
| 50 | 3 | 86.2 | 87.9 | 93.2 | 93.6 | 96.2 | 93.9 |
| 50 | 10 | 68.0 | 69.2 | 94.0 | 94.5 | 95.6 | 94.7 |
| 100 | 1 | 94.4 | 96.0 | 94.3 | 94.8 | 94.3 | 94.4 |
| 100 | 3 | 88.5 | 88.5 | 94.9 | 95.1 | 96.8 | 95.0 |
| 100 | 10 | 68.3 | 68.3 | 94.8 | 94.8 | 96.4 | 95.1 |
| 300 | 1 | 95.1 | 95.1 | 94.9 | 95.2 | 94.9 | 95.1 |
| 300 | 3 | 87.8 | 88.9 | 95.1 | 95.2 | 97.5 | 95.2 |
| 300 | 10 | 68.1 | 68.8 | 95.0 | 95.0 | 96.3 | 94.9 |

## E Local-model results

**Table 3.** Local models, 120 tasks × 10 runs. Score, pass$^3$, pass$^{10}$ in %; steps: mean agent turns; step-limit: share of runs ending at the step budget; right columns: success rate (%) per task family.

| Model | score | pass$^3$ | pass$^{10}$ | steps | step-limit | fs | sql | crm | booking | forms | tabular |
|---|---|---|---|---|---|---|---|---|---|---|---|
| qwen2.5-7b | 63.7 | 52.9 | 40.8 | 6.0 | 2% | 90 | 57 | 55 | 88 | 63 | 28 |
| qwen2.5-3b-q8 | 45.1 | 28.0 | 15.8 | 4.7 | 1% | 64 | 62 | 60 | 64 | 15 | 4 |
| llama3.2-3b | 33.6 | 18.8 | 10.8 | 8.5 | 22% | 26 | 24 | 30 | 74 | 39 | 8 |
| qwen2.5-3b | 31.8 | 15.2 | 10.0 | 4.8 | 6% | 56 | 25 | 36 | 52 | 12 | 10 |
| qwen2.5-1.5b | 13.6 | 5.3 | 1.7 | 2.9 | 0% | 28 | 12 | 3 | 34 | 3 | 0 |
| llama3.2-1b | 4.6 | 0.5 | 0.0 | 8.5 | 42% | 4 | 8 | 0 | 14 | 2 | 0 |
| qwen2.5-3b-q2 | 0.0 | 0.0 | 0.0 | 1.0 | 0% | 0 | 0 | 0 | 0 | 0 | 0 |

## F Final audit tables

**Monte Carlo precision.** Every simulated proportion is a Monte Carlo estimate. Table 4 gives the number of replications per cell and the worst-case standard error ($p = \frac{1}{2}$); the standard error at a reported value $\hat{p}$ is $\sqrt{\hat{p}(1-\hat{p})/\text{reps}}$, so coverage figures should not be read beyond about one decimal and differences of one or two points between methods are within simulation noise in the 400–1000-replication experiments.

**Table 4.** Replications per simulation cell and worst-case Monte Carlo standard error.

| Simulation | Replications per cell | Worst-case MC s.e. (points) |
|---|---|---|
| S0 interval coverage | 2000 | 1.1 |
| S0 type-I error | 2000 | 1.1 |
| S0 ranking calibration | 400 | 2.5 |
| S4 sequential | 600 | 2.0 |
| E1 skewed weak null | 1500 | 1.3 |
| E2 allocation | 1500 | 1.3 |
| E3 coverage robustness | 1000 | 1.6 |
| E5 sequential margins | 2000 | 1.1 |
| E6 pairwise Holm | 400 | 2.5 |
| Audit A equivalence | 2000 | 1.1 |

**Sequential equivalence.** Table 5 shows decision rates of the sequential procedure of Section 3 on a three-point law for the per-task difference, 2000 replications per row and at most 300 tasks, with Clopper–Pearson intervals for the rate of wrong decisions. "Wrong decision" counts $a$ better when $\Delta \le \varepsilon$, $b$ better when $\Delta \ge -\varepsilon$ and equivalent when $|\Delta| > \varepsilon$; "CS miss" is the rate at which the sequence excludes the true $\Delta$ at some task. The largest observed wrong-decision rate is 2.70% (95% upper limit 3.51%); the largest rate of false equivalence, with a true difference outside the band, is 0.55% (upper limit 0.98%). These are finite-sample frequencies under one family of laws, consistent with the bound $\alpha = 5\%$ but not a proof of it, and they are not themselves guarantees.

**$\tau$-bench sensitivity.** Table 6 applies every procedure to the same trajectories. All mean-score procedures use the 115 retail and 50 airline tasks with four trials; each cell gives the raw $p$-value and, in parentheses, the Holm-adjusted value over the two domains; the smallest attainable Monte Carlo permutation $p$-value is 1/100,001. Estimates: retail gap +10.2 points (task bootstrap [+3.9, +16.5]), airline gap +3.5 points [-6.0, +13.0]; the GEE intervals and odds ratios are in `results/audit_taubench.json`. The odds-ratio and majority rows estimate other quantities than the mean-score difference.

**SWE-bench Verified and missing logs.** Table 7 separates `no_generation` (failure) from `no_logs` (unobserved). "Least favourable filling" scores the missing cells of the lower-ranked entry of each pair as

**Table 5.** Sequential decisions by margin and true difference (all in percentage points of score, rates in %).

| Margin | True Δ | Region | $a$ (%) | $b$ (%) | Equiv. (%) | Undec. (%) | Wrong decision, 95% CP (%) | CS miss (%) |
|---|---|---|---|---|---|---|---|---|
| 0 | -2 | out | 0.3 | 2.6 | 0.0 | 97.1 | [0.11, 0.65] | 2.25 |
| 0 | +0 | in | 1.4 | 1.3 | 0.0 | 97.3 | [2.03, 3.51] | 2.70 |
| 0 | +2 | out | 3.2 | 0.6 | 0.0 | 96.2 | [0.31, 1.05] | 2.50 |
| 3 | -5 | out | 0.0 | 3.5 | 0.0 | 96.5 | [0.00, 0.18] | 2.50 |
| 3 | +0 | in | 0.2 | 0.1 | 0.0 | 99.6 | [0.17, 0.79] | 2.05 |
| 3 | +3 | in | 1.1 | 0.1 | 0.0 | 98.8 | [0.77, 1.78] | 2.65 |
| 3 | +5 | out | 3.4 | 0.0 | 0.0 | 96.6 | [0.00, 0.18] | 2.00 |
| 3 | +8 | out | 13.1 | 0.0 | 0.0 | 87.0 | [0.00, 0.18] | 2.80 |
| 6 | -8 | out | 0.0 | 2.5 | 0.0 | 97.5 | [0.00, 0.18] | 2.00 |
| 6 | +0 | in | 0.2 | 0.0 | 0.0 | 99.8 | [0.08, 0.58] | 2.55 |
| 6 | +6 | in | 1.0 | 0.0 | 0.0 | 99.0 | [0.61, 1.54] | 2.30 |
| 6 | +8 | out | 3.1 | 0.0 | 0.0 | 96.9 | [0.00, 0.18] | 1.95 |
| 6 | +14 | out | 39.7 | 0.0 | 0.0 | 60.3 | [0.00, 0.18] | 2.90 |
| 10 | -12 | out | 0.0 | 3.8 | 0.4 | 95.9 | [0.14, 0.72] | 2.65 |
| 10 | +0 | in | 0.0 | 0.1 | 9.2 | 90.7 | [0.00, 0.28] | 1.90 |
| 10 | +10 | in | 1.1 | 0.0 | 1.2 | 97.8 | [0.65, 1.60] | 2.30 |
| 10 | +12 | out | 3.1 | 0.0 | 0.5 | 96.4 | [0.27, 0.98] | 2.45 |
| 10 | +22 | out | 87.4 | 0.0 | 0.0 | 12.6 | [0.00, 0.18] | 2.10 |

resolved and those of the higher-ranked entry as unresolved, then applies Holm over all pairs; it is a bound for each pair separately, not a joint scenario.

## G Reproducibility status

Verification levels are reported separately (details and commands in `FINAL_REPRODUCIBILITY_REPORT.md`).

1. **Raw outcomes from recorded trajectories.** Possible for the two public datasets only: `scripts/fetch_taubench.py`, `fetch_swebench.py` and `fetch_swebench_meta.py` download the trajectories and result files, and the re-analyses (`reanalyze_*.py`, `audit_taubench.py`, `audit_swebench.py`) recompute every statistic from them. The SWE-bench data correspond to `swe-bench/experiments` in a state consistent with commit `92c5a4f` (inferred, not recorded at download; see Section 6.2). **Not possible** for the local-model study: raw trajectories, per-run outcomes and the record/replay cache are not in the repository.

2. **Statistics from raw outcomes.** Done for $\tau$-bench and SWE-bench (above; independent recomputation by the audit scripts, stored `reanalysis_*.json` not regenerated) and, in part, for the simulations, which are seeded scripts: the S0 components coverage, type-I error, design and ranking were rerun in the final audit and matched the stored first-version values bit for bit; the S0 sequential component (S4), E5 and the equivalence study were rerun after the sequential fix, so S4 and E5 no longer equal the first-version values (S4 was then rerun again and matched the current stored file bit for bit); E1, E2, E3 and E6 were not rerun in the audit (stored results only, produced once in the first revision). Not possible for the local-model study; the per-pair $p$-values stored in `results/real_main.json` allowed only the pair-count sensitivity to the degenerate model.

3. **Aggregates from stored aggregates.** Every table, figure and number is regenerated by `scripts/make_paper_assets.py` from `results/*.json`; this checks typesetting and consistency, not the experiments.

4. **Independent repetition of the evaluation.** Not performed for the local models (no Ollama server was available in the audit environment and the per-run outcomes needed for a comparison are not stored; the original run took hours on the author's laptop). Model tags and quantisations are in `configs/models.json`, the protocol in Appendix C; temperature 0.7, 256 generated tokens per turn, four workers and seed 0 are in `scripts/run_real.py`, but hardware, software versions

**Table 6.** Sensitivity of the $\tau$-bench retail and airline inference to the analysis choice (gpt-4o versus Claude 3.5 Sonnet; raw $p$-values, two-sided).

| Method | Estimand / null | Type | Retail $p$ | Airline $p$ | Limitation |
|---|---|---|---|---|---|
| Task bootstrap (20k) | mean-score difference; $H_0$: bootstrap re-centred | simulation | 0.002 (0.004) | 0.512 (0.512) | interval is percentile; needs enough tasks |
| Studentised sign-flip (primary) | mean-score difference; $E[d] = 0$ | permutation, MC (100k) | 0.002 (0.005) | 0.545 (0.545) | asymptotic for skewed $d$; one-sample theory not proved |
| Plain sign-flip | symmetric $d_t$ about 0 | permutation, MC (100k) | 0.002 (0.005) | 0.549 (0.549) | exact only under symmetry |
| Paired $t$ on task means | mean-score difference; $E[d] = 0$ | asymptotic | 0.002 (0.004) | 0.483 (0.483) | normal approximation, 115/50 tasks |
| Wilcoxon signed-rank | symmetric $d_t$; zeros dropped | asymptotic | 0.004 (0.008) | 0.449 (0.449) | drops tasks with $d_t = 0$; different null |
| Within-task label permutation | identical outcome law per task | permutation, MC (100k) | $2.0 \times 10^{-5}$ ($4.0 \times 10^{-5}$) | 0.449 (0.449) | sharp null, stronger than equal means |
| GEE, identity link, task clusters | mean-score difference | asymptotic (sandwich) | 0.002 (0.003) | 0.475 (0.475) | equals task-level mean test in large samples |
| GEE, logit link, task clusters | marginal odds ratio | asymptotic (sandwich) | 0.002 (0.004) | 0.475 (0.475) | different scale (odds ratio) |
| Logistic, task fixed effects | task-conditional odds ratio | asymptotic, cluster-robust | 0.004 (0.009) | 0.506 (0.506) | different estimand; many nuisance parameters |
| McNemar, majority (2/4 = fail) | difference of majority-success rates | exact binomial | 0.163 (0.326) | 0.267 (0.326) | arbitrary tie rule; run information lost |
| McNemar, majority, ties dropped | same, on non-tied tasks | exact binomial | 0.096 (0.141) | 0.070 (0.141) | changes the task population |
| McNemar, pass$^4$ | difference of all-4-succeed rates | exact binomial | 0.003 (0.007) | 1.000 (1.000) | reliability estimand |
| McNemar, pass@4 | difference of any-of-4 rates | exact binomial | 0.115 (0.231) | 1.000 (1.000) | ceiling-type estimand |
| Pooled $z$ (runs independent) | two proportions, runs as units | asymptotic | 0.001 (0.002) | 0.480 (0.480) | invalid: ignores clustering and pairing |

**Table 7.** SWE-bench Verified: distinguishability under alternative treatments of unobserved instances.

| Analysis | Entries | Pairs | Separable (%, Holm) | Adjacent, Holm | Adjacent, raw |
|---|---|---|---|---|---|
| All entries unobserved = unresolved (official score) | 129 | 8 256 | 71.7 | 0/128 | 2/128 |
| Entries with no missing logs | 88 | 3 828 | 72.3 | 0/87 | 1/87 |
| No missing logs one attempt declared | 27 | 351 | 59.3 | 4/26 | 7/26 |
| Public log folders cover 500 and no missing logs | 74 | 2 701 | 70.8 | 0/73 | 1/73 |
| All 129 pairwise complete cases | 129 | 8 256 | 71.4 | – | – |
| All 129 least favourable filling of missing logs | 129 | 8 256 | 70.8 | – | – |

and the Ollama build were not archived beyond what the text states. Server-side nondeterminism means a repetition would not reproduce individual outcomes even with the same seed. What was and was not repeated in the audit is itemised in item 2 and in `FINAL_REPRODUCIBILITY_REPORT.md`: S0 coverage, type-I error, design and ranking, bit for bit against the first-version values; S4, E5 and the equivalence study, after the sequential correction (S4 also against the current file); E1, E2, E3 and E6 not repeated (stored results of the first revision); the public SWE-bench and $\tau$-bench re-analyses recomputed independently from the raw public files by the audit scripts (headline values matched; the stored `reanalysis_*.json` files were not regenerated); the local-model study not reproducible, because its raw data are not in the repository.

**Tests.** The test suite (`pytest`, no network) passes, including audit tests for the betting capital process, the variance components and the allocation rule; `ruff` (check and format) is clean on the final tree; `mypy -strict` could not be run locally (a numpy stub rejects the local Python 3.14 interpreter); the continuous-integration results, including the type-check job, are those reported on the pull request, not in this paper. Seeds are fixed in every script. Compute: simulations on a laptop CPU in minutes to tens of minutes; no cloud and no paid API. Code, benchmark generator and scripts: `https://github.com/fragompul/agentaudit`.