AgentCert Bench: Measuring the Gap Between Agent-Reported and Independently Verified Success

Ziwei Guo
AgentCert
https://agentcert.app

Version 0.5, August 2026

Abstract

Agent benchmarks commonly report whether an agent appears to complete a task, while deployment decisions depend on whether the intended state change actually occurred, policy was respected, and evidence is complete. We introduce AgentCert Bench, a 24-task executable assurance benchmark spanning browser, coding, data, and messaging agents. Each run separates agent-reported success from an independent outcome probe and a digest-bound held-out evaluator. We execute a preregistered $2 \times 4 \times 2$ model–harness–control matrix with five independent repetitions per cell, comprising 480 task runs. Across all behavioral runs, apparent success was 91.9% [95% CI 89.1%–94.0%], independently verified success was 87.5% [95% CI 84.2%–90.2%], and the resulting Assurance Gap was 4.4% [95% CI 2.9%–5.6%]. The benchmark records signed evaluator receipts, policy violations, evidence completeness, token usage, cost, latency, and infrastructure errors. This repeated study remains deliberately small and synthetic; its contribution is a reproducible measurement boundary and multiplicity-aware paired analysis, not a general model ranking.

1 Introduction

Tool-using agents can produce convincing completion messages even when they changed the wrong record, violated a policy, failed to persist a result, or omitted evidence needed to establish what happened. Existing interactive benchmarks provide increasingly realistic environments [2, 3], but a single task-success score can collapse distinct questions: what the agent claimed, what the target system recorded, which controls constrained execution, and whether the result is auditable.

AgentCert Bench evaluates these questions separately. Its primary metric is

$$\text{Assurance Gap} = \Pr(\text{apparent success}) - \Pr(\text{verified success}). \tag{1}$$

The metric is paired at task level. Positive values indicate over-claiming; negative values indicate under-claiming or a state transition that occurred despite an unsuccessful final report. Neither direction is silently folded into ordinary task failure.

We ask three research questions:

1. How often does apparent task success differ from independently verified outcome success?
2. Do execution controls change verified success, policy violations, or the Assurance Gap?
3. What cost and latency accompany each model–harness combination?

2 Benchmark Design

2.1 Task packs

The benchmark contains six tasks in each of four capability packs. Browser tasks cover UI drift, blocking overlays, page-level prompt injection, approval, and false-success reporting. Coding tasks cover bounded

edits, secret boundaries, test-before-claim behavior, protected branches, retries, and truthful failure reporting. Data tasks cover read-only aggregation, scoped updates, destructive operations, redaction, transaction rollback, and count verification. Messaging tasks cover draft-only behavior, approval, recipient allowlists, secret exfiltration, idempotency, and delivery verification.

Every public task declares an instruction, policy constraints, expected state transition, evidence requirements, and resource budgets. The execution process never receives the held-out predicate or signing key. A private evaluator adds a per-task fixture marker, checks the independently observed state, and returns an Ed25519-signed receipt bound to the observation and predicate digests.

2.2 Agent and evaluator boundary

The four harnesses are Browser Use 0.13.7, a smolagents 1.26.0 coding tool agent, a smolagents 1.26.0 data agent, and a LangGraph 1.2.10 messaging agent. Agent bridges and outcome probes are separate processes. The probe reads local synthetic system state after execution and cannot change the Agent's final claim. The evaluator is a separate HTTP service with private predicates and signing credentials.

2.3 Controls

The baseline condition supplies policy text but does not enforce it outside the agent. The enforced condition applies deterministic gateway checks at the tool or action boundary, including least-privilege file access, scoped writes, approval requirements, allowlists, destructive-action blocks, retry limits, and transaction rollback. Independent outcome probes run in both conditions and are measurement infrastructure, not an experimental control.

3 Experimental Setup

We compare pinned snapshots `gpt-4.1-mini-2025-04-14` and `gpt-5-mini-2025-08-07`. Each compatible model–harness–control cell executes six tasks in five independent repetitions, yielding 16 cell families, 80 executions, and 480 task runs. The study lock records model IDs, harness versions, controls, evaluator key ID, token prices, and repetition seeds. Each task has a hard timeout, a single attempt, an observation-size cap, a token cap, and a dollar cap.

Binary rates use two-sided 95% Wilson score intervals. Run-to-run variability is reported as the sample standard deviation over five repetitions. Assurance Gap and effect intervals use a 10,000-sample repetition-cluster bootstrap. Control effects pair baseline and enforced outcomes by task and repetition; model effects use the same pairing. We report risk difference and matched odds ratio. Formal inference uses an exact sign-flip test over the five independent repetition-level mean effects and applies Holm correction separately to the eight control and eight model comparison families. Task-level exact McNemar results are retained only as descriptive diagnostics because repeated executions of the same six tasks are not independent. Infrastructure errors are reported separately and excluded from behavioral-rate denominators.

4 Results

The complete matrix contained 480 behavioral runs and 0 infrastructure errors. Total measured model cost was $2.4817 and aggregate agent latency was 8217.5 seconds. Because each repetition contains only six tasks, task coverage remains bounded even though run variance is now measurable. Mean verified-success run SD across cell families was 4.7%. The results support failure analysis within this benchmark revision but not broad claims about model superiority.

The aggregate signed gap should not be read as an error rate: 441 runs appeared successful and 420 were independently verified, producing the small net difference in 4.4% [95% CI 2.9%–5.6%]. At task level, however, 48 runs overclaimed success and 27 underclaimed it. Thus 75 of 480 runs (15.6%) had different apparent and verified outcomes. The mutually exclusive primary labels were 388 verified, 49 policy violation, 32 silent partial success, and 11 task failure. Signed gap, directional disagreement, policy compliance, and pass rate therefore answer different questions and are reported separately.

Model	Harness	Control	n	Apparent	Verified	Gap	Cost
gpt-4.1-mini-2025-04-14	browser-use@0.13.7	baseline	30/30	100.0%	96.7%	3.3%	$0.0721
gpt-4.1-mini-2025-04-14	browser-use@0.13.7	enforced	30/30	100.0%	96.7%	3.3%	$0.0738
gpt-4.1-mini-2025-04-14	langgraph@1.2.10	baseline	30/30	100.0%	83.3%	16.7%	$0.0026
gpt-4.1-mini-2025-04-14	langgraph@1.2.10	enforced	30/30	100.0%	83.3%	16.7%	$0.0026
gpt-4.1-mini-2025-04-14	smolagents-tool-coding@1.26.0	baseline	30/30	96.7%	96.7%	0.0%	$0.0198
gpt-4.1-mini-2025-04-14	smolagents-tool-coding@1.26.0	enforced	30/30	100.0%	100.0%	0.0%	$0.0198
gpt-4.1-mini-2025-04-14	smolagents@1.26.0	baseline	30/30	100.0%	60.0%	40.0%	$0.0119
gpt-4.1-mini-2025-04-14	smolagents@1.26.0	enforced	30/30	83.3%	66.7%	16.7%	$0.0130
gpt-5-mini-2025-08-07	browser-use@0.13.7	baseline	30/30	66.7%	100.0%	-33.3%	$0.0973
gpt-5-mini-2025-08-07	browser-use@0.13.7	enforced	30/30	40.0%	96.7%	-56.7%	$0.1113
gpt-5-mini-2025-08-07	langgraph@1.2.10	baseline	30/30	100.0%	93.3%	6.7%	$0.0102
gpt-5-mini-2025-08-07	langgraph@1.2.10	enforced	30/30	96.7%	83.3%	13.3%	$0.0088
gpt-5-mini-2025-08-07	smolagents-tool-coding@1.26.0	baseline	30/30	100.0%	100.0%	0.0%	$0.0154
gpt-5-mini-2025-08-07	smolagents-tool-coding@1.26.0	enforced	30/30	100.0%	96.7%	3.3%	$0.0156
gpt-5-mini-2025-08-07	smolagents@1.26.0	baseline	30/30	100.0%	66.7%	33.3%	$0.0106
gpt-5-mini-2025-08-07	smolagents@1.26.0	enforced	30/30	86.7%	80.0%	6.7%	$0.0115

Table 1: Model–harness–control results. Behavioral n excludes infrastructure errors; total n remains visible. Confidence intervals are retained in the machine-readable analysis and appendix to keep this table compact.

Across the eight task-and-repetition-paired baseline–enforced comparisons, verified success increased in 3, was unchanged in 2, and decreased in 3. After Holm correction, 0 of 8 comparisons rejected the no-change null at $\alpha = 0.05$. The controls therefore expose workflow-specific effects rather than establishing a universal benefit.

4.1 Failure classification

AgentCert assigns each task one mutually exclusive primary taxonomy label: verified, task failure, silent partial success, policy violation, insufficient evidence, or infrastructure error. State mismatches and policy identifiers remain available as multi-valued details. Table 1 reports verified outcomes while the reproduction archive contains task-level labels and signed receipts.

5 Related Work

WorkArena and BrowserGym evaluate web agents on enterprise knowledge-work tasks [2]; OSWorld and AndroidWorld extend interactive evaluation to desktop and mobile environments [5, 4]. AgencyBench studies long-horizon, multi-capability tasks and model–scaffold effects [3]. Our work is narrower: it instruments a cross-capability execution boundary and measures paired apparent versus independently observed outcomes. The NIST Generative AI Profile emphasizes lifecycle risk management and TEVV practices [1]. AgentCert operationalizes a small subset of that concern as executable evidence; it does not claim regulatory certification.

6 Limitations and Threats to Validity

The 24 tasks use deterministic local fixtures, not production systems. They represent four frameworks and two OpenAI model snapshots, not the agent ecosystem. Five repetitions estimate run-to-run variability for this locked setup but do not address task-selection uncertainty or provider drift. With only five independent repetition clusters, the minimum attainable two-sided exact sign-flip p-value is 0.0625; this release is therefore powered to estimate variance and effect size, not to reject effects at $\alpha = 0.05$, even before Holm correction. GPT-4.1 mini receives preregistered provider seeds; the pinned GPT-5 mini harness performs independent invocations but does not expose deterministic provider seed support. The evaluator is held out from the agent process but operated by the benchmark author, so it is not an independent third-party replication.

Control conditions alter tool enforcement and can also influence agent trajectories. Token cost uses published list prices and excludes infrastructure, cached-token discounts, and engineering cost.

The reference adapter is part of the trusted computing base. The orchestrator removes the private variant seed before launching Agent and probe subprocesses, but supplies a task-specific fixture marker needed to initialize the synthetic target system. Evaluator receipts attest which predicate scored which observation; they do not prove that a malicious adapter reported the underlying system state truthfully. Production source authenticity requires a customer-owned signed recorder or an independently controlled outcome probe.

7 Reproducibility and Ethics

The public reproduction bundle includes task manifests, adapters, lock files, analysis code, signature verification, and exact commands. Private predicates and signing keys are intentionally excluded; their digests and public verification key are published. All targets are synthetic local systems. No real email, payment, customer account, or production repository is modified. The benchmark should be used to diagnose scoped behavior, not to label an agent or vendor as generally safe or unsafe.

8 Conclusion

AgentCert Bench makes an agent's claim, the observed outcome, policy compliance, and evidence completeness independently visible. The repeated study demonstrates a reproducible path from real framework execution to signed held-out receipts, run-level variance, effect sizes, and multiplicity-aware paired comparisons. Larger task sets, additional providers, and third-party evaluator operation are required before leaderboard or certification claims are justified.

References

[1] Chloe Autio, Reva Schwartz, Jesse Dunietz, et al. Artificial intelligence risk management framework: Generative artificial intelligence profile. Technical Report NIST AI 600-1, National Institute of Standards and Technology, 2024.

[2] Alexandre Drouin, Maxime Gasse, Massimo Caccia, et al. Workarena: How capable are web agents at solving common knowledge work tasks? *arXiv preprint arXiv:2403.07718*, 2024.

[3] Keyu Li, Junhao Shi, Yang Xiao, et al. Agencybench: Benchmarking the frontiers of autonomous agents in 1m-token real-world contexts. *arXiv preprint arXiv:2601.11044*, 2026.

[4] Christopher Rawles, Alice Clinckemaillie, Yifan Chang, et al. Androidworld: A dynamic benchmarking environment for autonomous agents. *arXiv preprint arXiv:2405.14573*, 2024.

[5] Tianbao Xie, Danyang Zhang, Jixuan Chen, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *arXiv preprint arXiv:2404.07972*, 2024.

A Reproduction Appendix

A.1 Environment

The authoritative `study.lock.json` is checked into the 24-task study example directory. Each release retains the public source revision, operating system, Python executable, package locks, model snapshots, repetition seeds, evaluator public-key digest, and output manifest digests.

A.2 Execution

An operator starts the private evaluator with a predicate directory, Ed25519 private key, API token, service ID, and key ID. Agent framework environments are selected by explicit Python paths. The exact study command is:

```
python examples/agentcert-bench-24-task-study/run_study.py \
  --evaluator-url http://127.0.0.1:8091 \
  --repetitions 5 \
  --output .agentcert/bench-study-v0.5
```

A.3 Artifact verification

Each repetition-level cell contains `summary.json`, `results.jsonl`, JUnit, a replay manifest, normalized observations, traces, capability-specific artifacts, and signed evaluator receipts. The top-level directory contains matrix, analysis, per-family variance, and Markdown reports. Reproduction verifies every receipt against the release public key and every file against its manifest before comparing scores.

A.4 Statistical procedure

For a binary rate with x successes in n behavioral runs, we report a two-sided 95% Wilson score interval. Repetition-level rates report sample standard deviation and a t interval for the mean. For Assurance Gap and paired effects, complete repetitions are resampled as clusters 10,000 times. Control comparisons use enforced-minus-baseline outcomes paired by task and repetition; model comparisons use the same pairing. Risk difference and matched odds ratio are reported with an exact sign-flip test over repetition-level mean effects. Task-level exact McNemar results are descriptive only because tasks repeat across runs. Holm correction controls family-wise error separately across the eight control and eight model comparisons. Infrastructure errors are excluded from behavioral denominators and listed separately. With five repetition clusters, the minimum attainable two-sided exact sign-flip p-value is 0.0625. Corrected tests in this release are therefore descriptive power diagnostics rather than evidence of statistical rejection.

A.5 Required independent replication

An independent replication should operate its own evaluator service and signing key, regenerate private fixture markers, rerun all 16 cell families and five repetitions from a clean clone, and publish task-level receipts plus environment hashes. Agreement on aggregate rates without receipt and artifact verification is not considered a complete replication.