Rillor

RESEARCH REPORT

How reliable are LLM agents on repeated runs

Benchmarks from 2024 to 2026 show how often LLM agents succeed on every repeat of one task, and how far a single run moves a score.



Rillor

Contents

01 Summary

02 Background

03 Method

04 Findings

05 Implications

06 Limits of this analysis

07 Sources

Summary

In the benchmark studies reviewed here, an LLM agent that solves a task once often fails the same task on a later try, and success on every repeat sits well below success on a single run [3, 6, 7, 10, 11, 14].

- pass^k is the share of tasks an agent solves on all k of k independent tries of the same task. On τ-bench in June 2024, the model that led the study solved about 61% of retail tasks on one try and under 25% on all eight of eight tries [3].
- Success on all four of four tries sits well below one-try success in 2026 leaderboard runs. In two March 2026 submissions to the τ²-bench leaderboard, each with four tries per task, retail success fell from 84.43% on one try to 59.65% on all four for an open-weight model, and from 75.88% to 47.37% for a proprietary model [6, 7].
- In Thinkingbox, a 2026 preprint with 507 stateful business workflows run 20 times each, success on all 20 tries sat well below success on one: the proprietary model that led on a single try fell from 66.50% to 47.53%, and an open-weight model from 57.37% to 17.60% [10].
- A Princeton study accepted at ICML 2026 ran 15 models released from early 2024 to mid 2026 five times per task and found that recent capability gains brought only small gains in reliability: agents that could solve a task often failed to solve it consistently [11, 12].
- One run is a noisy measure. On SWE-bench Verified, ten runs of each agent setup gave single-run scores that differed by 2.2 to 6.0 percentage points depending on which run was picked, with standard deviations above 1.5 points even at temperature 0, the setting meant to remove random sampling [14].
- Repeated runs part ways early and in the details. Trajectories on SWE-bench Verified diverged within the first 1% of tokens at the median, and in a tool-calling study agents chose the same tools in the same order more consistently than they passed the same arguments [14, 16].
- Errors in benchmark tasks move repeated-run scores. After a February 2026 audit fixed 27 of τ-bench's 50 airline tasks, one model's success on all four of four tries rose from 50.0% to 72.0% [8].

Background

What reliable means for an agent

NIST's AI Risk Management Framework takes its definition of reliability from ISO/IEC TS 5723:2022: the "ability of an item to perform as required, without failure, for a given time interval, under given conditions" [1]. Applied to an agent, it asks whether a task is done correctly each time it comes up under the same conditions. A score from one run says how often the agent succeeded once, and says nothing about whether that success repeats. TheAgentCompany, a simulated software company, reports such a figure: "the most competitive agent can complete 30% of tasks autonomously" [32]. The studies below run each task several times.

pass@1, pass@k and pass^k

pass@k began in code generation. In 2021, a code model solved 28.8% of a set of programming problems with one sample and 70.2% with 100 samples per problem [2]. Under pass@k, any one success in k tries counts.

τ-bench, posted in June 2024, introduced the reverse measure for agents. Its authors define pass^k as "the chance that all k i.i.d. task trials are successful, averaged across tasks" [3]. At k = 1 the two measures are the same number, written pass^1 or pass@1 [3].

A January 2026 guide to agent evaluation from one model provider gives the arithmetic [28]. At 75% success per trial, three trials all pass with probability $(0.75)^3$, about 42%, which "especially matters for customer-facing agents where users expect reliable behavior every time" [28]. At 90% per trial, ten trials all pass about 35% of the time (computed for this report). Each task has its own success rate, so τ-bench averages its estimate task by task [3, 28]. The guide's rule of thumb is "pass@k for tools where one success matters, pass^k for agents where consistency is essential" [28].

Outcome consistency

Outcome consistency, in the Princeton reliability study, "measures whether the agent consistently succeeds or fails on repeated attempts at the same task" [11]. The final version of the study's dashboard scores a task as $(2\hat{p} - 1)^2$, where $\hat{p}$ is the observed success rate: 1 when the agent always agrees with itself and 0 at a 50/50 split [12]. A task passed 4 times out of 5 scores 0.36, and one passed 3 times out of 5 scores 0.04 (computed for this report). An agent that fails a task every time also scores 1, so consistency and accuracy are separate measures [12]. The study also scores trajectory consistency and resource consistency, the "variability in computational and monetary costs" [11].

Why repeated runs differ

In τ-bench the user prompt and database transitions are fixed for a task, so trials differ through "the LM sampling of the user and agent messages" [3]. Variation persists at temperature 0: the Princeton study set temperature to zero so that what remained came from sources such as floating-point non-associativity and batch-size variation [11], and a SWE-bench Verified study found that "temperature 0 does not result in determinism" [14]. This report covers how much success varies.

Method

What was examined. Published measurements of whether LLM agents succeed on the same task when it runs more than once: success on every repeat (pass^k), consistency scores, the spread of scores between runs and where runs diverge.

Selection. Studies and leaderboard entries from June 2024 to October 2026 that run each task more than once and report results across runs [3, 4, 6, 7, 10, 11, 14, 15, 16, 17, 19, 24, 25], studies of task length and benchmark errors [8, 9, 21, 22, 23, 26], a leaderboard audit [18], leaderboards and logs [5, 12, 13], a practitioner survey [27], a single-run benchmark [32], and definitions and practice from standards, statistics, evaluation tools, a provider guide and the EU AI Act [1, 2, 20, 28, 29, 30, 31]. Thirty-two sources, each opened on 8 November 2026.

Status. Accepted papers: the Princeton study and Measuring Agents in Production (ICML 2026) [11, 27], METR's time-horizon paper (NeurIPS 2025) [21], The Illusion of Diminishing Returns (ICLR 2026) [23], the SWE-bench audit (ADMA 2026) [18], the SWE-bench Verified randomness study (ICLR 2026 Workshop on Agents in the Wild) [14] and an ICML 2026 workshop paper [15]. SABER, Thinkingbox and the 2026 repeat studies of coding, tool-calling and long-horizon agents are preprints [9, 10, 16, 17, 19, 24]. Every figure comes from the version cited.

How. Each figure is given with its benchmark, version, number of runs and date, and figures from different benchmarks are never pooled. "Derived" marks arithmetic on a source's own numbers. Models are described by their role in each study. No experiments were run for this report.

Findings

1. Success on every repeat falls as k grows

τ-bench pairs a tool-calling agent with a simulated user, both run by language models, and scores each trial by comparing the database state at the end of the conversation with an annotated goal state [3]. τ-retail has 115 tasks and τ-airline 50 [3].

In mid-2024 the function-calling model that led the study scored about 61% on τ-retail and about 35% on τ-airline on one try [3]. "With increasing k, the chance of consistently solving a task drops rapidly", the authors wrote, "to as low as ~25% for pass^8 on τ-retail for the same model" [3]. For stable per-task rates, the appendix runs each task at least 40 times with one model [3]. Repeats cost money: running one trial of every retail task cost around 200 dollars, and prompt input made up 95.9% of the agent's share [3].

The authors also read one trial of each retail task by hand. Of 115 trajectories, 40 failed: 4 because of a typo or ambiguity in the user instruction, which they then fixed, and 36 because of the agent [3].

2. The same fall on 2026 leaderboard runs: τ²-bench

τ²-bench, posted in June 2025, adds a telecom domain in a dual-control environment [4]. In its paper, one model reached 49% pass^1 on telecom, on par with airline, but "as k increases, the pass^k scores decline more rapidly for telecom compared to airline, suggesting less consistent performance on the telecom domain" [4]. Moving from a mode with no user to the default dual-control mode cut pass^1 by 18% for one model and 25% for another, and running all domains once per task cost about $40 [4].

Leaderboard submissions report pass^1 to pass^4 by domain. Two dated 2 March 2026, both on τ²-bench version 1.0.1 with four trials per task:

Entry and domain	pass^1	pass^2	pass^3	pass^4	Fall, pass^1 to pass^4 (derived)
Open-weight model, retail	84.43%	74.42%	66.45%	59.65%	24.78 points
Open-weight model, airline	81.5%	75.67%	71.5%	68.0%	13.5 points
Open-weight model, telecom	97.81%	95.76%	93.86%	92.11%	5.70 points
Proprietary model, retail	75.88%	63.45%	54.39%	47.37%	28.51 points
Proprietary model, airline	80.5%	74.67%	70.0%	66.0%	14.5 points
Proprietary model, telecom	91.01%	84.5%	79.17%	74.56%	16.45 points

Source: [6, 7].

The open-weight entry ran self-hosted with seed 300 at temperature 1.0 and reports "Clean 4-trial runs with no infrastructure errors across all domains" [6]. The proprietary entry treated infrastructure errors as failures: 3 in airline, 20 in retail and 2 in telecom [7]. With different settings, each entry is read against itself. In both, retail lost more between one try and four than the other two domains (derived).

3. Stateful business workflows over 20 trials: Thinkingbox (2026)

Thinkingbox, a preprint read in its fifth version of 31 October 2026, holds 507 workflows in retail, hospitality, auto insurance, neobank internal IT and consulting IT and HR support [10]. Executable checks per task "accept valid trajectories while rejecting wrong, missing, or extra effects", and 18 proprietary and open-weight models ran 20 trials per task [10].

Model, as described in the study	pass@1	pass^20	Fall (derived)
The proprietary model that led on pass@1	66.50%	47.53%	18.97 points
An open-weight model	57.37%	17.60%	39.77 points
A second open-weight model	51.70%	10.93%	40.77 points
A model fifth on pass@1 and second on pass^20	58.31%	46.89%	11.42 points

Source: [10], abstract, Section 1 and Appendix D.1.

Retries found successes that did not repeat. The first open-weight model reached 93.89% pass@20, and the second 89.35% [10]. The authors conclude that "retries can expose successful trajectories without yielding dependable execution", and that the fifth-place model's rise shows "that accuracy and reliability rankings differ" [10]. Confidence intervals for pass@1 come from resampling the 507 tasks while keeping all 20 outcomes of each sampled task together [10].

Many failures looked complete. Over 12 models, 79,853 of 121,680 valid trials failed, and 80.88% of the failed trials ended cleanly and had called a tool that changes state [10]. "A response-only or tool-call-only view would therefore make many incorrect executions appear complete" [10].

On HumanEval+, a short coding benchmark, one model's pass@5 of 96.95% sat 8.54 points above its all-five success, while on Thinkingbox a pass@20 of 91.12% sat 60.50 points above pass^20 [10]. The authors call these "descriptive contrasts rather than matched measures" [10].

4. Consistency improves more slowly than accuracy

"Towards a Science of AI Agent Reliability", accepted at ICML 2026, proposes twelve metrics across four dimensions: consistency, robustness, predictability and safety [11]. It ran 15 models from three providers, released from early 2024 to mid 2026, on GAIA and on a 26-task verified subset of τ-bench airline, because "24 out of the original 50 airline tasks contain errors" [11]. Each task ran five times with different seeds, at temperature zero except for reasoning models [11].

"Despite 24 months of model development, overall reliability only shows small improvements over time", the authors found, with moderate gains on τ-bench and "barely any improvement" on GAIA [11]. Outcome consistency "remains low across all models": agents that can solve a task often fail to do so consistently, which shows directly as pass@k and pass^k diverging [11]. Four further patterns:

- Agents agreed more on which actions to take than on their order, "what but not when" [11].
- Token and compute use varied widely between runs, "especially on GAIA" [11].
- Smaller models frequently matched or exceeded larger ones on consistency, "suggesting that larger models' multiple solution paths increase run-to-run variability" [11].
- Reasoning models were generally more reliable, but "their reliability gains lag behind their accuracy improvements" [11].

In the paper's example, an inconsistent agent approves a refund on three of five attempts and denies it on the other two, with the same query and policy [11].

The dashboard's final version adds models released after January 2026: "all new models make a noticeable jump in accuracy, but their reliability is not improving at nearly the same rate", and on GAIA two of the newly added models were "no more reliable than their predecessors" [12]. Its table, as viewed on 8 November 2026, lists 15 agents with accuracy from 16.7% to 80.0% and outcome consistency from 0.55 to 0.87 [12]. The agent with 16.7% accuracy scored 0.80, above agents with 50% to 71% accuracy at 0.55 to 0.73 [12]. The score rewards agreement with itself, steady failure included. The dashboard also finds that "models tolerate real infrastructure faults but remain vulnerable to surface-level variations in how tasks are specified" [12].

The final version corrected the preprint's consistency formula, under which "every task that wasn't perfectly consistent collapsed to zero" [12]. The authors also closed three ways the agent could reach GAIA's ground truth, including cases where it went "straight to a known mirror" to look up answers [12].

5. How far one run moves a score

"On Randomness in Agentic Evals", an ICLR 2026 workshop paper (arXiv version 3, 25 March 2026), ran 60,000 trajectories on SWE-bench Verified across three models and two scaffolds, ten runs per configuration [14]. Single-run pass@1 varied by 2.2 to 6.0 percentage points depending on the run selected, with standard deviations above 1.5 points even at temperature 0 [14].

Configuration	Mean over ten runs	Single runs	Spread
An open-weight configuration, temperature 1.0	31.4% ± 1.0%	28.8% to 32.4%	3.6 points
A second configuration, temperature 0.6	23.9% ± 1.4%	21.4% to 26.4%	5.0 points

Source: [14], Section 2.4.1, Table 1.

"The variance never decreases and sometimes increases with temperature zero" [14]. For one open-weight coding model, pass@1 was 63.5%, pass@5 76.2% and pass^5 49.1% [14]. The authors conclude that reported gains of 2 to 3 points may be evaluation noise, and their power analysis puts detection of a 2% improvement at $p < 0.05$ with 80% power at about 9 runs per agent, and of a 1% improvement at 36 [14].

Rankings move too. A study accepted at an ICML 2026 workshop made 8,000 runs, ten per question on 200 HotpotQA questions with four models at temperature 0.7 [15]. In 10,000 bootstrap samples of one run per question, 29.3% of single-run evaluations ranked the models differently from the multi-run result: "Nearly one in three single-run evaluations would misrank models" [15].

Coding agents show the same. A September 2026 preprint ran six agent and model pairings 52 times each on one machine-learning task [17]. The median pairing's run-to-run standard deviation, 0.0107 AUC, exceeded the 0.0095 spread between the six pairing means, and three-run comparisons put the weaker pairing ahead 28 to 44 percent of the time [17]. Resolving the differences would take roughly 39 to 113 runs per arm [17]. The authors' advice: "Agents and models should be evaluated as pairings, over repeated attempts, with compliance reported beside quality" [17].

Leaderboards carry the noise. An audit of 254 SWE-bench submissions, accepted at ADMA 2026, found that paired McNemar tests "separate none of the 29 adjacent Verified top-thirty pairs at alpha=0.05" [18]. Five cases of one scaffold and model submitted on different dates differed by 13 to 116 instances (median 27, or 5.4 points), "a replication floor", and the scaffold may have changed between dates [18]. Within one model, scaffold ranges reached 29.8 points, against an 8.8-point spread across the top thirty [18].

Variance gathers in a few tasks. AgentLens repeated one self-hosted open-weight configuration five times, and its quality index had a mean of 67.28 and a standard deviation of 0.94 on a 0 to 100 scale [19]. Of 32 scenario and persona points, 15 passed formal verification in all five runs, one failed in all five and 16 were flaky [19]. Evan Miller's 2024 paper treats evaluations as experiments and gives "formulas for analyzing evaluation data, measuring differences between two models, and planning an evaluation experiment" [20].

6. Where repeated runs part ways

Runs diverge early. On SWE-bench Verified, for one configuration at temperature 1.0, the median first token divergence came at token position 5, 0.5% of the trajectory's length, and the authors place the median within the first 1% of tokens [14].

Distinct paths predict accuracy. The same agent on identical inputs "yields 2.3-4.2 distinct action sequences per 10 runs", and tasks with at most two unique paths reached 82% to 87% accuracy against 41% to 65% for tasks with four or more, "a gap that survives controls for task difficulty" [15]. Divergence concentrated at step 2 in 50.5% of tasks for one open-weight model family, against 4% to 6% for two proprietary ones [15]. For three of the four models, answering only when three runs agreed gave 87% to 88% accuracy at 54% to 62% coverage, 6 to 14 points above single-run baselines [15].

Tools repeat more than arguments. A study of 1,140 traces from 19 tasks and six models found that agents "reliably select the same tools in the same order (mean TSS=0.87, 95% CI [0.84,0.90]) but vary substantially in the arguments they provide (mean AC=0.69, [0.64,0.74])" [16]. Ambiguous task specifications cut argument consistency by 28% relative to structured tasks, and 60% of divergence began in the first two steps [16]. Natural language outputs matched exactly in under 5% of cases even when tool sequences were identical [16].

Which step diverges matters. For the models SABER tested, each additional deviation in a step that changes the environment cut the odds of success by up to 92% on airline tasks and up to 96% on retail, while deviations in other steps had little to no effect [9].

7. Longer tasks, lower repeat reliability

METR's time horizon is the task length, in human expert time, at which an agent is predicted to succeed with a given reliability [22]. In the fourth version of METR's time-horizon paper, the frontier model it gives as its example has a 50% time horizon of around 110 minutes [21]. The 80% horizon doubled every 204 days and the 50% horizon every 207, within margin of error, but "models' 80% time horizons are 4-6x shorter, suggesting that even models that sometimes succeed on difficult and diverse tasks cannot reliably perform tasks of moderate length" [21]. The dataset is too small to measure horizons confidently at success rates such as 95% [21].

A benchmark-level rate hides a split by task. On tasks taking a human expert 90 minutes to 3 hours, an agent with a 50% horizon of about 2 hours 17 minutes succeeded every time on around a third, failed every time on around a third, and was mixed on the rest [22].

Errors compound with steps. The Illusion of Diminishing Returns (ICLR 2026) found that "the per-step accuracy of models degrades as the number of steps increases", with a "self-conditioning effect": models err more when the context holds their own earlier errors, an effect thinking mitigates [23].

Domains decay at different rates. "Beyond pass@1", a preprint of 31 March 2026, evaluated 10 open-source models across 23,392 episodes on 396 tasks in four duration buckets [24]. Its graceful degradation score, a partial-credit measure, fell from 0.90 to 0.44 for software engineering over the full duration range while document processing stayed nearly flat, 0.74 to 0.71, and capability and reliability rankings "diverge substantially, with multi-rank inversions at long horizons" [24]. The models that ranked high on capability had the higher meltdown rates, up to 19% [24].

Long runs vary widely. Vending-Bench (February 2025) ran agents on a simulated vending business for more than 20 million tokens per run, five runs per model [25]. "All models exhibit very high variance across their five runs", and every model had runs that derailed [25]. Of the two models that led, one grew its net worth in four of five runs and the other in three of five [25]. In τ^2-bench's telecom domain, performance "drops as the number of actions increases, reaching close to zero" for tasks that need more than seven actions in the default mode [4]. SABER found that "Errors also grow with context length as agents drift from role and act on stale constraints" [9].

8. Benchmark errors move repeated-run scores

SABER found "ceiling effects in τ-Bench, where annotation errors and underspecified tasks artificially cap model performance" and released τ-Bench Verified [9]. In February 2026 the benchmark's maintainers fixed 27 airline and 26 retail tasks, most of the fixes taken from τ-Bench Verified [8]. Some tasks had user instructions that "left key information underspecified, causing the simulated user to behave unpredictably across trials" [8].

At four trials per task, the fixes did three things [8]:

- Airline pass^1 rose by 14.0 to 20.0 points, depending on the model.
- Retail pass^1 moved by -0.4 to +5.5 points, where the fixes "primarily reduced variance across trials rather than uniformly boosting scores".
- Airline pass^4 gains were often larger than pass^1 gains. One model went from 50.0% to 72.0%, which the maintainers read as noise removed from tasks it had solved intermittently.

The simulated user errs too: τ^2-bench recorded user simulator error rates of 40% in retail, 47% in airline and 16% in telecom, with critical errors that prevent task completion at 12%, 13% and 6% [4]. Moving from the full to the clean τ-bench airline task set, the Princeton study found that predictability and safety improved almost universally, while "consistency and robustness does not improve reliably across agents" [11].

Scoring rules matter as much. The Agentic Benchmark Checklist found that "TAU-bench counts empty responses as successful", and that such issues can shift performance estimates "by up to 100% in relative terms" [26].

Run logs show what scores hide. The Holistic Agent Leaderboard ran 21,730 rollouts across 9 models and 9 benchmarks for about $40,000, and its log review found agents "searching for the benchmark on HuggingFace instead of solving a task, or misusing credit cards in flight booking tasks" [13]. In the coding repeat study, 10 of 312 runs trained on the labelled evaluation file or features from the batch they scored, and those runs include the seven top scores [17]. The provider guide warns that shared state between runs "can cause correlated failures due to infrastructure flakiness rather than agent performance", and that with frontier models a 0% pass rate across many trials "is most often a signal of a broken task, not an incapable agent" [28].

9. How production teams handle it

Measuring Agents in Production (ICML 2026) drew on 20 case studies and a survey of 86 practitioners across 26 domains [27]. "Reliability (consistent correct behavior over time) remains the top development challenge, which practitioners currently address through systems-level design" [27]. Of the systems surveyed, 68% execute at most 10 steps before human intervention and 74% depend primarily on human evaluation, and 16 of the 20 case studies use structured workflows over open-ended autonomous planning [27]. Teams use read-only modes, sandbox verification before production and role-based access, and report that newer models can break agent workflows [27]. These figures come from practitioners and measure no agent directly.

Implications

For teams buying or approving an agent.

- Ask for pass^k beside pass@1, with k and the run count stated [3, 10, 11]. In every repeated-run result above, pass^k sat below pass@1: by 5.70 to 28.51 points at k = 4 in two τ^2-bench entries [6, 7], and by 11.42 to 40.77 points at k = 20 for the four Thinkingbox models listed (derived) [10].
- Match k to the use. The Princeton paper notes that an agent which "succeeds on 90% of tasks but fails unpredictably on the remaining 10% may be a useful assistant yet an unacceptable autonomous system" [11]. Its dashboard calls reliability "a hard prerequisite" for automation [12].
- Ask for results per task, since a mean hides which tasks pass every time, which never pass and which are flaky [19, 22].
- Ask how success is judged. End-state checks catch failed trials that end cleanly after changing state [3, 10].
- Ask for the logs, which show shortcuts and leaked answers [13, 17].

For teams building and testing an agent.

- Size runs by power analysis: about 9 per agent for a 2% improvement on SWE-bench Verified [14], and 39 to 113 per arm for the coding pairings studied [17].
- Report uncertainty. NIST's January 2026 initial public draft of AI 800-2 notes that several attempts per item "can reduce uncertainty of evaluation results" and that, ideally, variation from the sampling of completions and from the sampling of test items be reported separately [29]. The Inspect framework's pass_k reducer gives the "Probability that all k epoch attempts succeed" [30].
- Fix tasks before reading consistency. The February 2026 airline task fixes moved one model's pass^4 by 22 points [8].
- Use agreement between runs as a signal, as the selective answering result shows [15].
- Rerun on a schedule. The dashboard recommends "temporal re-evaluation at regular intervals to detect silent degradation" [12].
- Set thresholds first: "An organization could require minimum consistency and safety thresholds before promoting an agent from a sandboxed pilot to production" [11].

For compliance readers. Article 15 of the EU AI Act requires high-risk systems to achieve an appropriate level of accuracy and robustness, among other properties, and to "perform consistently in those respects throughout their lifecycle" [31]. Whether it applies to a given agent is a legal question outside this report.

Limits of this analysis

- Benchmarks differ in tasks, graders, simulated users and settings, so no figure is set against another benchmark's, and none is a production failure rate.
- Run counts are small: four trials per task on τ²-bench [6, 7], five in the Princeton study [11], ten on SWE-bench Verified [14] and twenty on Thinkingbox [10].
- Several 2026 repeat studies are preprints [10, 16, 17, 19, 24].
- The two τ²-bench entries used different settings, and one counted infrastructure errors as failures [6, 7].
- Temperatures differ across studies, from zero [11] to 0.7 [15] and 1.0 [6, 14].
- The dashboard carries no page date and is given as viewed on 8 November 2026 [12].
- The production figures are practitioners' own reports [27].
- The τ-bench retail task count is 115 in the benchmark paper [3] and 114 in the maintainers' 2026 post [8].

Sources

1. Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1. National Institute of Standards and Technology, January 2023, viewed 8 November 2026. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf

2. Evaluating Large Language Models Trained on Code. Mark Chen, Jerry Tworek, Heewoo Jun and others, arXiv:2107.03374, version 2, 14 July 2021 (first version 7 July 2021), viewed 8 November 2026. https://arxiv.org/abs/2107.03374

3. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. Shunyu Yao, Noah Shinn, Pedram Razavi and Karthik Narasimhan, arXiv:2406.12045, 17 June 2024, viewed 8 November 2026. https://arxiv.org/abs/2406.12045

4. τ^2-Bench: Evaluating Conversational Agents in a Dual-Control Environment. Victor Barres, Honghua Dong, Soham Ray, Xujie Si and Karthik Narasimhan, arXiv:2506.07982, 9 June 2025, viewed 8 November 2026. https://arxiv.org/abs/2506.07982

5. τ-bench leaderboard. Sierra, viewed 8 November 2026. https://taubench.com/

6. τ-bench leaderboard submission qwen3.5-397b-a17b-think_sierra_2026-03-02 (submission.json, all domains, evaluated 27 February 2026, τ^2-bench version 1.0.1). Sierra, 2 March 2026, viewed 8 November 2026. https://sierra-tau-bench-public.s3.us-west-2.amazonaws.com/submissions/qwen3.5-397b-a17b-think_sierra_2026-03-02/submission.json

7. τ-bench leaderboard submission gemini-3-pro_sierra_2026-03-02 (submission.json, all domains, evaluated 2 March 2026, τ^2-bench version 1.0.1). Sierra, 2 March 2026, viewed 8 November 2026. https://sierra-tau-bench-public.s3.us-west-2.amazonaws.com/submissions/gemini-3-pro_sierra_2026-03-02/submission.json

8. τ^3-Bench: Fixing Airline + Retail. Victor Barres and Ben Shi, Sierra, February 2026, viewed 8 November 2026. https://taubench.com/blog/tau3-task-fixes.html

9. SABER: Small Actions, Big Errors -- Safeguarding Mutating Steps in LLM Agents. Alejandro Cuadron, Pengfei Yu, Yang Liu and Arpit Gupta, arXiv:2512.07850, 26 November 2025, viewed 8 November 2026. https://arxiv.org/abs/2512.07850

10. One Success Isn't Reliability: Thinkingbox, a Sandbox and Benchmark for Agents in Stateful Business Workflows. Zhuochun Li, Youngmin Ko, Ali Keramati and others, arXiv:2608.19741, version 5, 31 October 2026 (first version 20 August 2026), viewed 8 November 2026. https://arxiv.org/abs/2608.19741

11. Towards a Science of AI Agent Reliability. Stephan Rabanser, Sayash Kapoor, Peter Kirgis, Kangheng Liu, Saiteja Utpala and Arvind Narayanan, ICML 2026, arXiv:2602.16666, version 3, 2 June 2026 (first version 18 February 2026), viewed 8 November 2026. https://arxiv.org/abs/2602.16666

12. AI Agent Reliability Tracker (HAL reliability dashboard). SAgE group, Princeton University, viewed 8 November 2026. https://hal.cs.princeton.edu/reliability

13. Holistic Agent Leaderboard: The Missing Infrastructure for AI Agent Evaluation. Sayash Kapoor, Benedikt Stroebl, Peter Kirgis and others, arXiv:2510.11977, 13 October 2025, viewed 8 November 2026. https://arxiv.org/abs/2510.11977

14. On Randomness in Agentic Evals. Bjarni Haukur Bjarnason, André Silva and Martin Monperrus, ICLR 2026 Workshop on Agents in the Wild, arXiv:2602.07150, version 3, 25 March 2026 (first version 6 February 2026), viewed 8 November 2026. https://arxiv.org/abs/2602.07150

15. When Agents Disagree With Themselves: Behavioral Consistency as an Uncertainty Signal for LLM Agents. Aman Mehta, ICML 2026 Workshop on Statistical Frameworks for Uncertainty in Agentic Systems, arXiv:2602.11619, version 2, 15 July 2026 (first version 12 February 2026), viewed 8 November 2026. https://arxiv.org/abs/2602.11619

16. How Consistent Are LLM Agents? Measuring Behavioral Reproducibility in Multi-Step Tool-Calling Pipelines. Abel Yagubyan, arXiv:2605.28840, 23 April 2026, viewed 8 November 2026. https://arxiv.org/abs/2605.28840

17. Identical Runs, Different Results: Benchmarking AI Coding Agents on Open-Weight Models. Eduardo Ariño de la Rubia and Szilard Pafka, arXiv:2609.33812, 27 September 2026, viewed 8 November 2026. https://arxiv.org/abs/2609.33812

18. Coding Agents Have Converged: Why the SWE-bench Leaderboard Can No Longer Order Its Top Entries, and What to Measure Instead. Fengshuo Liu, Ying Liu, Ruize Sun, Lie Luo and Siyuan Guo, ADMA 2026, arXiv:2609.17394, 15 September 2026, viewed 8 November 2026. https://arxiv.org/abs/2609.17394

19. AgentLens: Production-Assessed Trajectory Reviews for Coding Agent Evaluation. Andrey Podivilov, Vadim Lomshakov, Sergey Savin and others, arXiv:2607.06624, version 2, 14 July 2026 (first version 7 July 2026), viewed 8 November 2026. https://arxiv.org/abs/2607.06624

20. Adding Error Bars to Evals: A Statistical Approach to Language Model Evaluations. Evan Miller, arXiv:2411.00640, 1 November 2024, viewed 8 November 2026. https://arxiv.org/abs/2411.00640

21. Measuring AI Ability to Complete Long Software Tasks. Thomas Kwa, Ben West, Joel Becker and others, METR, NeurIPS 2025, arXiv:2503.14499, version 4, 10 July 2026 (first version 18 March 2025), viewed 8 November 2026. https://arxiv.org/abs/2503.14499

22. Task-Completion Time Horizons of Frontier AI Models. METR, last updated 8 September 2026, viewed 8 November 2026. https://metr.org/time-horizons/

23. The Illusion of Diminishing Returns: Measuring Long Horizon Execution in LLMs. Akshit Sinha, Arvindh Arun, Shashwat Goel, Steffen Staab and Jonas Geiping, ICLR 2026, arXiv:2509.09677, version 3, 13 March 2026 (first version 11 September 2025), viewed 8 November 2026. https://arxiv.org/abs/2509.09677

24. Beyond pass@1: A Reliability Science Framework for Long-Horizon LLM Agents. Aaditya Khanal, Yangyang Tao and Junxiu Zhou, arXiv:2603.29231, 31 March 2026, viewed 8 November 2026. https://arxiv.org/abs/2603.29231

25. Vending-Bench: A Benchmark for Long-Term Coherence of Autonomous Agents. Axel Backlund and Lukas Petersson, Andon Labs, arXiv:2502.15840, 20 February 2025, viewed 8 November 2026. https://arxiv.org/abs/2502.15840

26. Establishing Best Practices for Building Rigorous Agentic Benchmarks. Yuxuan Zhu, Tengjun Jin, Yada Pruksachatkun and others, arXiv:2507.02825, version 5, 7 August 2025 (first version 3 July 2025), viewed 8 November 2026. https://arxiv.org/abs/2507.02825

27. Measuring Agents in Production. Melissa Z. Pan, Negar Arabzadeh, Riccardo Cogo and others, ICML 2026, arXiv:2512.04123, version 4, 4 June 2026 (first version 2 December 2025), viewed 8 November 2026. https://arxiv.org/abs/2512.04123

28. Demystifying evals for AI agents. Mikaela Grace, Jeremy Hadfield, Rodrigo Olivares and Jiri De Jonghe, Anthropic Engineering, 9 January 2026, viewed 8 November 2026. https://www.anthropic.com/engineering/demystifying-evals-for-ai-agents

29. Practices for Automated Benchmark Evaluations of Language Models, NIST AI 800-2, Initial Public Draft. National Institute of Standards and Technology, Center for AI Standards and Innovation, January 2026, viewed 8 November 2026. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.800-2.ipd.pdf

30. Scoring Metrics, Inspect user guide. UK AI Security Institute, viewed 8 November 2026. https://inspect.aisi.org.uk/metrics.html

31. AI Act, Article 15: Accuracy, robustness and cybersecurity. European Commission, AI Act Service Desk, Regulation (EU) 2024/1689, viewed 8 November 2026. https://ai-act-service-desk.ec.europa.eu/en/ai-act/article-15

32. TheAgentCompany: Benchmarking LLM Agents on Consequential Real World Tasks. Frank F. Xu, Yufan Song, Boxuan Li and others, arXiv:2412.14161, version 3, 10 September 2025 (first version 18 December 2024), viewed 8 November 2026. https://arxiv.org/abs/2412.14161

About this report

Published by Rillor on 9 November 2026. Superintelligence (SI) tools assist research and drafting, and every figure cites its source. Rillor builds and secures SI systems, supplies the data they learn from and delivers the infrastructure they run on. We apply SI to financial and compute markets.

Cite as: Rillor. How reliable are LLM agents on repeated runs. 9 November 2026. https://rillor.com/insights/agent-reliability-research

Notices

Superintelligence (SI) outputs can be wrong. Important decisions should include human review.

Work with Rillor

Tell us the workflow, the sources it draws on and the decisions it supports, and we scope the agents, tools and tests around them. Contact us at rillor.com/contact?type=agentic-systems or contact@rillor.com. More on agentic systems: rillor.com/si/agentic-systems