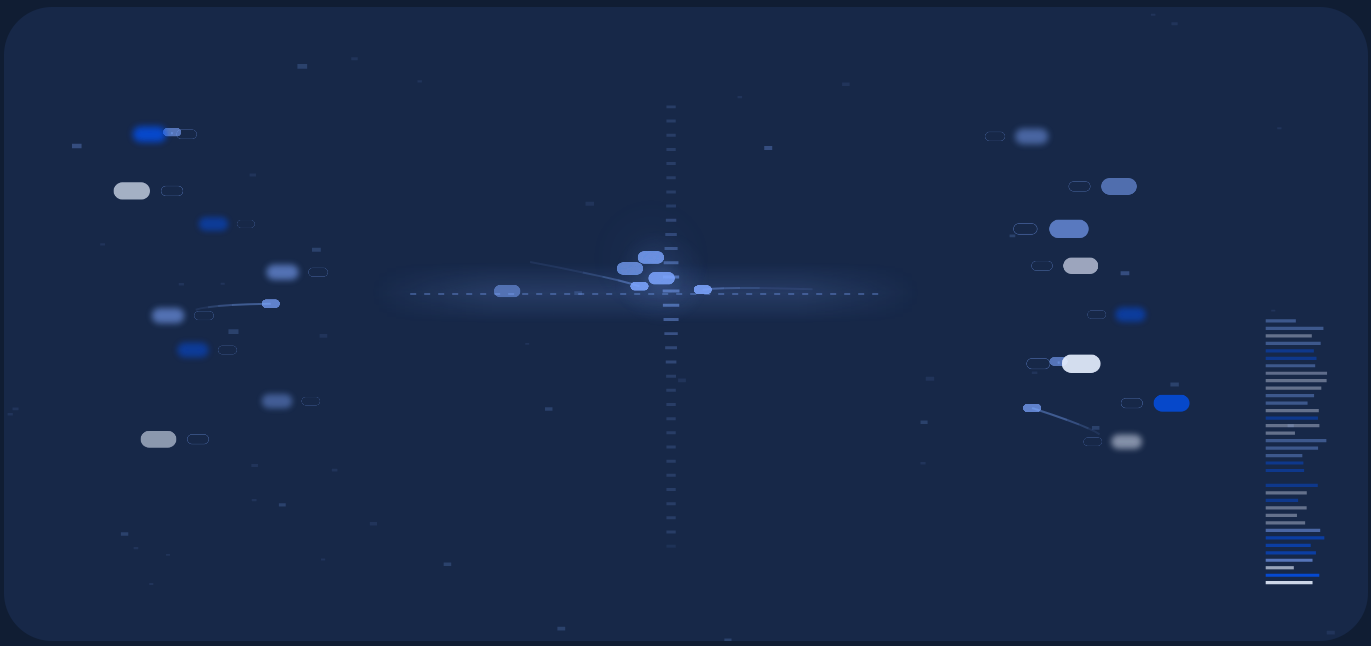
Rillor

RESEARCH REPORT

What it costs to evaluate an agent

Agent evaluation cost in published research: what single benchmark runs cost, and how repeated trials, sample size, judges, people and smaller task sets change the total.



Rillor

Contents

01 Summary

02 Background

03 Method

04 Findings

05 Implications

06 Limits of this analysis

07 Sources

Summary

Published research puts one agent benchmark run anywhere from 12 cents to $2,829, and the number of trials, tasks, graders and people decides what an evaluation program costs in total [3, 4, 5].

- On the Holistic Agent Leaderboard (HAL), the reported cost of one full run of one agent on one benchmark ranged from $0.12 to $2,829 in April 2026, four orders of magnitude, and three orders within some individual benchmarks [5]. The $0.12 run scored 1.54% and the $2,829 run 28.48% [3, 4].
- HAL's dollar figures are reported values. A known-issue note dated 5 October 2026 on its leaderboard pages says costs and token counts are overstated, by exactly 2 times for affected models and unevenly across rows, and that caching is not counted [3, 4].
- Repeated trials multiply every cost line. The metric pass^k, the chance that all k trials of a task succeed, needs at least k runs of each task [12], and the EvalEval Coalition estimates that 8 reruns would take HAL's roughly $40,000 of rollouts to about $320,000 [1, 5].
- Smaller task sets save less on agents than on static tests: 100 to 200 times less compute for static benchmarks against 2 to 3.5 times for agentic ones, according to the EvalEval Coalition [5].
- Grading and human baselines are cost lines of their own. A model judge cost about $66 per paper on PaperBench at March 2025 prices, against tens of hours of expert grading per paper [13], and on RE-Bench, METR's benchmark of 8-hour research engineering tasks, human experts were paid about $1,855 per attempt against about $123 per agent run [15].
- Three model providers list evaluations as a use for batch processing at 50% of standard prices [20, 21, 22]. The Inspect framework's documentation says agentic tasks in batch mode can take many hours or days, so the discount suits judge calls and single-turn items [23].
- In illustrative arithmetic from published SWE-bench costs, one 500-task run at the median $0.344 per task comes to about $172, eight trials to about $1,376 and eight trials on three model versions to about $4,128 [29].

Background

A static language model benchmark sends each test item once and scores the answer. At scale that was already costly. In HELM's evaluation, as reported in the paper's October 2023 version, 4,939 runs used 12,169,227,491 tokens and 17,431,479 queries, at $38,001 for commercial APIs and about 19,500 GPU hours for the open models [9]. The tinyBenchmarks authors, citing HELM's authors, put one model's HELM evaluation at over 4,000 GPU hours, or over $10,000 for APIs [7]. Perlitz and colleagues describe costs of thousands of GPU hours per model for such benchmarks [8].

An agent evaluation adds steps and an environment. Each task runs as a loop of model calls and tool calls inside a repository, a browser, a desktop or a simulated conversation, until the agent stops or reaches a limit. HAL's authors note that tasks need very different resources, from basic CPU to high-memory GPU instances, and that serial execution of agent evaluations can take weeks [1].

The authors of "AI Agents That Matter" split the cost of an agent into fixed costs, paid once to tune its hyperparameters, and variable costs, paid each time it runs [10]. On HumanEval they found that, for substantially similar accuracy, cost could differ by almost two orders of magnitude [10]. They ask that downstream evaluations report input and output token counts next to dollar costs, because "the cost of an API call might change overnight" [10].

Standards and leaderboards now treat cost as part of the result. NIST's January 2026 initial public draft, AI 800-2, lists the human labor for setup and running and the computational cost among its practical considerations [18]. It calls cost control an emerging practice and asks evaluators to report costs alongside performance [18]. HAL, run by a Princeton team, starts from the view that "lack of cost tracking prevents cost-controlled evaluations" and shows cost next to accuracy on its boards [2].

Method

What was examined. Published evidence on the cost of running agent evaluations: per-run benchmark costs, what drives their spread, repeated trials, sample size, smaller task sets, batch and caching discounts for evaluation traffic, and the cost of graders and human baselines. The production cost of one finished task is a separate question and is left out.

Sources. HAL's paper and leaderboard pages [1, 2, 3, 4] and the EvalEval Coalition's April 2026 analysis of HAL data [5]. Benchmark papers with cost sections: HELM [9], tau-bench [12], PaperBench [13], MLE-bench [14], RE-Bench [15] and HCAST [16]. Methods papers on compression and statistics [6, 7, 8, 11], AI Agents That Matter [10] and NIST AI 800-2 [18]. Public leaderboard data from SWE-bench and BFCL [27, 28, 29, 30, 31, 35]. The UK AI Security Institute's Inspect documentation [23, 24, 25, 26] and its March 2026 study of token budgets [33]. Batch and pricing pages from three model providers [19, 20, 21, 22], one model developer's guide to agent evals [32], AlpacaEval [17] and Epoch AI's price data [34].

How. Figures come from the cited passage, table or data file, each with its date and setup. Arithmetic done for this report is labelled and rests on cited inputs. The SWE-bench and BFCL figures are computed from their published data, using every entry that carries a cost. Models are described by tier, and providers are not named in the text. The report uses no Rillor data.

HAL's cost warning. HAL's leaderboard pages carry a known-issue note dated 5 October 2026: reported costs and token counts are overstated [3, 4]. Nested trace spans counted the same API response twice, so affected models are inflated by exactly 2 times, models logged as a single span are unaffected, and the figures have not been re-derived [3, 4]. The pages also say that costs leave out caching [3, 4]. The HAL paper's total [1], the EvalEval ranges [5] and Ndzomga's medians [6] come from the same harness or leaderboard. Every HAL-derived dollar figure here is therefore a reported value that may be up to twice the API cost, by an amount that varies across rows.

Findings

1. Published per-run costs

HAL's paper reports 21,730 agent rollouts across 9 models and 9 benchmarks in coding, web navigation, science and customer service, at a total cost of about $40,000 and prices as of 24 September 2025 [1]. That is about $1.84 per rollout and, from the 2.5 billion tokens of logged model calls, about 115,000 tokens per rollout (arithmetic on [1]). By benchmark, ScienceAgentBench averaged $13 per evaluation and Online Mind2Web over $450, and the team skipped one top-tier model on Online Mind2Web because it estimated the runs at about $20,000 [1].

By April 2026 the leaderboard held 26,597 rollouts [5]. The EvalEval Coalition read the cost of one run, one agent across all of a benchmark's tasks, from the live board:

Benchmark	Tasks	Reported cost per run, low to high	Median per run
TAU-bench Airline	50	$0.31 to $180	$22
CORE-Bench Hard	45	$2 to $510	$66
SciCode	65	$0.12 to $625	$67
GAIA	165	$7.80 to $2,829	$140
SWE-bench Verified Mini	50	$4 to $1,600	$163
Online Mind2Web	300	$5 to $1,610	$276

Sources: ranges [5], task counts and medians [6]. ScienceAgentBench ran from $0.19 to $77 [5].

The spread is four orders of magnitude across HAL's benchmarks and three within some individual benchmarks [5]. Ndzomga, working from the same leaderboard, reports per-task costs on SWE-bench Verified from $0.08 to $32.00 and a spread of 400 times driven by model pricing and scaffold design [6]. He finds that evaluating one agent on all eight benchmarks he selected costs a median of roughly $800 in API fees [6]. His total across 242 agent runs, about $46,000, is a tally of HAL's reported costs [6].

The ends of the range are different agents. The $0.12 SciCode run scored 1.54% [4]. The $2,829 GAIA run scored 28.48%, and EvalEval notes the figure is before caching [3, 5]. On Online Mind2Web one agent cost $171 and another $1,577, a ninefold gap, while the cheaper one scored 42.3% against 40.0% [1].

Boards that fix more of the setup show narrower ranges. SWE-bench's bash-only Verified board runs every model in the same mini-SWE-agent environment on 500 instances [29]. Across its 44 runs with costs, dated 20 July 2025 to 26 February 2026, the average cost per instance ran from $0.028 to $1.53, with a median of $0.344, and whole runs from $14.04 to $765.57 (computed from the board's data [29]). BFCL V4, updated 12 April 2026, gives an estimated cost for the entire benchmark for each model [27]. Its 109 entries run from $0.46 to $355.17, with a median of $18.25, and the top-ranked entry, at 77.47% overall, is listed at $86.55 [28].

Long-horizon benchmarks cost more per run. PaperBench's authors put one run of their agent at about $400 in API credits per paper, $8,000 for its 20 papers, plus $66 per paper for grading [13]. One run of MLE-bench's main setup, 24 hours on each of 75 competitions, needs 1,800 GPU hours, and one agent used 127.5 million input and 15.0 million output tokens per seed [14].

2. What drives the spread

Token prices. At HAL's September 2025 prices, a top-tier model cost $15 per million input tokens and $75 per million output tokens, against $0.10 and $0.40 for a small model, two orders of magnitude apart [1].

Scaffolds and reasoning settings. The scaffold decides how many calls an agent makes. On HumanEval, Reflexion and LDB cost over 50% more than the authors' warming strategy, and LATS over 50 times more [10]. HAL found that higher reasoning effort reduced accuracy in the majority of runs, and that the run with the top cost on a benchmark sat on the Pareto frontier in only 1 of 9 benchmarks [1].

Input tokens. In tau-bench, input made up 95.9% of the agent's price, mainly a long system prompt of domain policy and function definitions [12]. The agent and the simulated user cost $0.38 and $0.23 per task in mid-2024 [12]. RE-Bench agents used about 29 million input tokens and 499,000 output tokens per 8-hour run, about $123, and the authors note that the cost could potentially be much lower with proper prompt caching [15].

Limits. Step, token and cost limits cap the tail. The mini-SWE-agent configuration for SWE-bench sets a step limit of 250 and a cost limit of $3 per instance on its current main branch [31], and no single task in the board's data passes $3.80 [29]. Inspect lets an evaluator cap time, messages, tokens, turns or cost per sample, with an example limit of $2.00 per sample, and scores whatever output exists when a limit is reached [25]. Limits trade cost against measured capability. On its cyber tasks, the UK AI Security Institute found that recent models can productively use token budgets 10 to 50 times larger than typical evaluation settings allow, and about 8% of its tasks were solved only when the budget rose from 10 to 50 million tokens [33]. That finding covers those tasks only. NIST AI 800-2 notes that performance can often be raised by running a system in higher-execution-cost modes [18].

Environments. HAL's harness runs evaluations in parallel across hundreds of virtual machines, which cut evaluation time from weeks to hours, and the team built the orchestration from scratch as a significant engineering project [1]. SWE-bench recommends at least 120GB of free storage, 16GB of RAM and 8 CPU

cores [35]. MLE-bench gives each run 36 vCPUs, 440GB of RAM and one A10 GPU for up to 24 hours [14]. RE-Bench environments have 20 to 48 vCPUs, 200 to 400 GB of RAM and up to 6 H100s, and its $123 per run leaves the H100s out [15].

Spend limits and failures. HAL found one provider's default spend limit was $5,000 per month even at its top spending tier, so larger runs needed special approval, and parallel runs hit rate limits at every provider [1]. Agents that hit rate limits without retries failed silently and were scored as wrong answers [1]. In failed tasks on three benchmarks, agents met an environmental barrier, such as a crashing browser or an unavailable file, in roughly 40% of cases [1].

3. The cost of repeated trials

tau-bench defines pass^k as the chance that all k independent trials of a task succeed, averaged across tasks, and estimates it from n trials per task, so it needs at least k runs of every task [12]. Its authors ran at least 3 trials per task [12]. Their function-calling agent on a top-tier model succeeded on over 60% of tasks on average, and its pass^8 fell below 25% [12].

Each trial repeats the full cost of a run. EvalEval estimates that a HAL-style evaluation with 8 reruns per cell takes the roughly $40,000 aggregate to about $320,000, and calls reliability a multiplier on every cost category [5]. That figure is their estimate. HAL has since paused adding new models to focus on measuring reliability [2].

Published practice varies. AI Agents That Matter evaluated each agent five times on the test set [10]. MLE-bench repeated all experiments with 3 seeds and ran 16 and 36 seeds for two configurations [14]. NIST AI 800-2 notes that more trials per item, called epochs in the Inspect framework, raise cost but can reduce uncertainty, a tradeoff with the evaluation budget [18]. Inspect sets epochs per task, with an example of 5 epochs over 100 samples [26].

Repeats also shrink error, at a falling rate. In Miller's analysis of binary scores with uniform difficulty, going from one answer per question to two cuts total variance by a third, four answers by half and six by five ninths, with a ceiling of two thirds [11]. In his worked example with 198 questions, raising samples per question from 1 to 10 lowered the minimum detectable effect from 13.2% to 7.5% [11]. Each extra sample costs one more full pass, so variance falls by less per dollar as samples rise (arithmetic on [11]).

4. Sample size and statistical power

The task count an evaluation needs depends on the difference it has to detect. Miller's example, with fictional but reasonable parameters, needs at least about 969 independent questions to detect a 3-point difference with 80% power and a 5% false positive rate, and he suggests new evals contain at least 1,000 questions [11]. He recommends power analysis to judge whether an eval, or a random subsample, can test the hypothesis of interest [11]. With a fixed number of questions, he writes, a user can calculate the minimum detectable effect and decide whether the eval is worth running [11].

Large differences need far fewer tasks. One model developer's engineering guide says 20 to 50 simple tasks drawn from real failures are a good start, because large effects need small samples, and that more mature agents may need larger evals to detect smaller ones [32]. NIST AI 800-2 says the number of test items should balance statistical power against the budget [18]. tau-bench's authors write that a small set of high-quality tasks run for multiple trials can reveal rich insights [12].

The sample size multiplies the per-task cost. At the SWE-bench median of $0.344 per instance [29], Miller's 969 questions would cost about $333 per trial, against about $17 for a 50-task set (arithmetic).

5. Smaller task sets that keep rankings

For static benchmarks, curated subsets keep rankings at a fraction of the cost. tinyBenchmarks found 100 curated examples enough to estimate performance on MMLU's 14,000, a 140-fold reduction, and 30 examples per scenario enough for the Open LLM Leaderboard, a 160-fold reduction [7]. Estimation error stayed within 2% and never passed 4%, except for one model with extremely low performance [7]. Perlitz and colleagues report reductions in computation, often of 100 times or more, on HELM with minimal loss of reliability, and show that a benchmark leader can change when a low-ranked model is removed [8].

Agentic benchmarks compress less. The EvalEval Coalition puts the evidence at 100 to 200 times for static benchmarks, 2 to 3.5 times for agentic ones and about 1 times for evaluations with training in the loop [5]. Ndzomga's method, tested across eight benchmarks, 33 agent scaffolds and over 70 model configurations, runs a new agent only on tasks with historical pass rates between 30% and 70% [6]. It cut task counts by 44% to 70%, a median of 58%, and from 1,006 to 373 tasks across the benchmarks, for a median saving of $446.7 per agent, assuming cost scales linearly with task count [6]. Savings ran from $5 to $22 on tau-bench Airline to $149 to $253 on Online Mind2Web [6].

The method predicts rank order, which stayed stable under scaffold and temporal shifts, while predictions of absolute scores degraded [6]. It gave more reliable rankings than random sampling, whose results varied widely across seeds [6]. SciCode had too few mid-range tasks to filter, and the author calls the 30% to 70% band a practical heuristic [6].

Maintainers publish smaller official sets. SWE-bench Lite holds 300 tasks, selected to preserve the original's distribution and difficulty and to reduce evaluation costs [30]. HAL used SWE-bench Verified Mini, 50 of the 500 Verified tasks, which it notes is cheaper to evaluate [1]. MLE-bench recommends its Low complexity split of 22 competitions when running all splits is too resource-intensive [14]. PaperBench's authors expect Code-Dev runs to take half as long as full runs, for about $4,000 per run [13]. Its grading costs about 85% less, and its scores are only weakly correlated with the full benchmark [13].

6. Batch and caching discounts

Three providers sell asynchronous batch processing at 50% of standard prices and name evaluations as a use [20, 21, 22]. Their terms differ. One targets a 24-hour turnaround, with up to 50,000 requests per batch [21]. Another says batches usually finish in under an hour, expires any batch not done within 24 hours and caps

each at 100,000 requests or 256 MB [20].

Batch fits some evaluation traffic better than other traffic. Inspect's documentation says evaluations with few sequential generations, such as a question-answer eval with a model scorer, are a good fit [23]. Agentic tasks, with many generations that depend on each other, can take many hours or days in batch mode [23]. With sandboxes, the machine's resources can cap the number of samples that run at once [23]. One provider's managed agent sessions have no batch mode [19].

Caching lowers the cost of repeated input. On one provider's rate card, viewed 13 November 2026, a cache read costs 0.1 times the base input price on most models and 0.05 or 0.025 times on six others, including the $2-per-million mid-tier model used in section 8 [19]. Batch and caching discounts can be combined [19]. In a batch, the same provider gives no guarantee of cache hits, and users typically see hit rates from 30% to 98% [20]. HAL's figures leave caching out [1, 3]. Inspect can also cache model output on the evaluator's side, which saves calls while an evaluator iterates on a scorer. With an explicitly versioned model name, a new model version is called fresh instead of being served from the cache [24].

List prices move. Epoch AI found that the price of reaching a fixed level of performance fell by 9 to 900 times per year depending on the milestone, with a median of 50 times, and cautioned that the recent pace of decline may not persist [34]. Every dollar figure in this report carries its price date for that reason.

7. People and judges

Grading is a cost line of its own. PaperBench's rubrics took multiple weeks per paper to build with the papers' authors, and manual grading by experts took on the order of tens of hours per paper [13]. Its model judge cost about $66 per paper at March 2025 prices, from about 50 million input and 2 million output tokens, and about $10 per paper for Code-Dev [13]. A larger judge scored an F1 of 0.84 against 0.83 for about $830 per paper [13]. In preliminary results, an experimental judge with pruned rubrics cut grading costs tenfold [13].

For short outputs the gap between people and models is wider. AlpacaEval lists $300 per 1,000 examples for crowd workers paid $21 an hour, with human agreement of 65.7 [17]. Its model annotators run from $0.4 per 1,000, at agreement of 67.5, to $15.3, and one at $13.6 reached 69.2 [17]. The maintainers recommend validating automatic evaluators on one's own instructions and human annotations [17].

One model developer's guide rates model-based graders as more expensive than code and in need of calibration with human graders, and human studies as relatively expensive, slow and hard to run frequently [32]. NIST AI 800-2 says evaluators should consider checking LLM judges of transcripts against human judges' decisions [18]. A judge therefore carries the cost of the human labels used to check it [17, 18, 32].

Human baselines cost more again. HCAST paid baseliners $50 to $100 per hour plus $25 to $150 per hour in performance bonuses, for 563 baselines totalling over 1,500 hours, and 61% of those baselines succeeded [16]. At the base rates alone, 1,500 hours comes to $75,000 to $150,000 before bonuses (arithmetic on [16]).

RE-Bench paid its human experts about $1,855 per 8-hour attempt on average, against about $123 in tokens per agent run [15].

8. A worked budget from public prices

The table is illustrative arithmetic from cited figures. Each line keeps its source's price date.

Line	Inputs	Result
One SWE-bench Verified run at the median published cost	500 tasks at $0.344, runs dated July 2025 to February 2026 [29]	About $172 (whole runs $14.04 to $765.57 [29])
Eight trials per task, enough for pass^8	Line 1 times 8 [12]	About $1,376 ($112 to $6,125 across published runs)
Three model versions compared	Line 2 times 3	About $4,128
Own 50-task set, eight trials, conversational agent	50 times 8 times ($0.38 plus $0.23), mid-2024 [12]	$244
One 8-hour agent run at a mid-tier list price	29M input at $2 and 0.499M output at $10 per million tokens [15, 19]	About $63, or about $7.89 if every input token were a cache hit at that model's listed $0.10 per million [19]
One MLE-bench seed at the same list price	127.5M input at $2 and 15.0M output at $10 per million [14, 19]	About $405, plus 1,800 GPU hours [14]
Grading one PaperBench paper	$66 at March 2025 prices [13], at 50% if the judge runs in batch [21]	$33 to $66
Ten expert baseline attempts	10 times $1,855 [15]	About $18,550

The list prices on lines 5 and 6 are one provider's, viewed 13 November 2026 [19]. The all-cache-hit figure is a floor for comparison. Batch pricing does not suit the multi-turn runs on lines 5 and 6 [19, 23]. Lines 1 to 4 cover API fees only. Environments, GPU hours, engineering time and the human labels a judge needs come on top [1, 14, 18].

Implications

For evaluation leads budgeting a program. The cost of an evaluation multiplies per-task cost by tasks, trials and model versions, then adds grading, people and environments [5, 11, 13, 16]. The order of decisions follows from that. Decide what size of difference matters, set the task count from it with a power calculation [11], then choose the number of trials. A rerun grid is where budgets grow: eight trials multiply API fees eight times [5, 12]. Infrastructure failures need watching, since agents that hit rate limits were scored as wrong answers [1]. Report cost and token counts next to accuracy, as AI Agents That Matter and NIST AI 800-2 both ask [10, 18].

For buyers reading cost claims. A cost per run holds only for its scaffold, limits, price date and caching treatment [1, 3, 25, 33]. A low cost can mean a weak agent: the $0.12 run on HAL scored 1.54% [4]. HAL's own figures carry its October 2026 warning [3, 4].

For teams cutting cost. Each lever has published evidence and a tradeoff. Mid-difficulty task filtering cut task counts 44% to 70% and kept rank order, while its predictions of absolute scores degraded [6]. Official subsets such as SWE-bench Lite exist for cheaper runs [30]. PaperBench's cheaper variant correlates only weakly with the full benchmark [13]. Batch rates fit judge calls and single-turn items [20, 23], caching fits shared prompts [19, 24], and per-sample cost limits cap the tail [25]. Tighter budgets can lower measured capability, as the AI Security Institute found on its tasks [33].

Rillor's agent evaluation (https://rillor.com/si/evaluation#outputs) work delivers benchmark suites: fixed sets of tasks and markets with scoring code, rerun on each new model version so results stay comparable over time.

Limits of this analysis

- HAL-derived figures are overstated by up to 2 times for affected models, unevenly across rows, leave out caching and have not been re-derived [3, 4]. HAL has paused adding new models [2], so its figures cover models up to that pause.
- The $320,000 is the EvalEval Coalition's estimate [5]. Ndzomga's savings assume cost scales linearly with task count [6].
- Price dates run from the HELM paper of 2022 and 2023 to November 2026 [9, 19], and the price of a fixed level of performance has fallen 9 to 900 times per year [34], so figures from different dates are not added together.
- SWE-bench and BFCL figures are computed from data read on 8 November 2026, entries with costs only [28, 29]. The mini-SWE-agent configuration read is the current main branch, and older runs may have used other limits [31].
- tau-bench's per-trial total in its text does not reconcile with its per-task figures, so only the per-task figures are used [12].
- The AI Security Institute's budget findings cover its cyber tasks only [33].
- No source read prices the virtual machines or GPUs for evaluation environments, so infrastructure is stated as hardware and GPU hours.
- Engineering time for harnesses, tasks and scoring code is priced by no source read. HAL describes its orchestration as a significant engineering project [1], and PaperBench's rubrics took weeks per paper [13].
- Models are described by tier, and providers are not named in the text.

Sources

1. 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

2. HAL: Holistic Agent Leaderboard. SAgE team, Princeton University, viewed 8 November 2026. https://hal.cs.princeton.edu/

3. GAIA Leaderboard. HAL, Princeton University, viewed 8 November 2026. https://hal.cs.princeton.edu/gaia

4. Scicode Leaderboard. HAL, Princeton University, viewed 8 November 2026. https://hal.cs.princeton.edu/scicode

5. AI evals are becoming the new compute bottleneck. Avijit Ghosh, Yifan Mai, Georgia Channing and Leshem Choshen, EvalEval Coalition, 29 April 2026, viewed 8 November 2026. https://evalevalai.com/research/2026/04/29/eval-costs-bottleneck/

6. Efficient Benchmarking of AI Agents. Franck Ndzomga, arXiv:2603.23749, 24 March 2026, viewed 8 November 2026. https://arxiv.org/abs/2603.23749

7. tinyBenchmarks: evaluating LLMs with fewer examples. Felipe Maia Polo, Lucas Weber, Leshem Choshen and others, ICML 2024, arXiv:2402.14992, version 2, 26 May 2024, viewed 8 November 2026. https://arxiv.org/abs/2402.14992

8. Efficient Benchmarking of Language Models. Yotam Perlitz, Elron Bandel, Ariel Gera and others, NAACL 2024, arXiv:2308.11696, version 5, 1 April 2024, viewed 8 November 2026. https://arxiv.org/abs/2308.11696

9. Holistic Evaluation of Language Models. Percy Liang, Rishi Bommasani, Tony Lee and others, arXiv:2211.09110, version 2, 1 October 2023, viewed 8 November 2026. https://arxiv.org/abs/2211.09110

10. AI Agents That Matter. Sayash Kapoor, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir and Arvind Narayanan, arXiv:2407.01502, 1 July 2024, viewed 8 November 2026. https://arxiv.org/abs/2407.01502

11. 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

12. 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

13. PaperBench: Evaluating AI's Ability to Replicate AI Research. Giulio Starace, Oliver Jaffe, Dane Sherburn and others, arXiv:2504.01848, version 3, 7 April 2025, viewed 8 November 2026. https://arxiv.org/abs/2504.01848

14. MLE-bench: Evaluating Machine Learning Agents on Machine Learning Engineering. Jun Shern Chan, Neil Chowdhury, Oliver Jaffe and others, arXiv:2410.07095, version 6, 26 February 2025, viewed 8 November 2026. https://arxiv.org/abs/2410.07095

15. RE-Bench: Evaluating frontier AI R&D capabilities of language model agents against human experts. Hjalmar Wijk, Tao Lin, Joel Becker and others, METR, arXiv:2411.15114, version 2, 27 May 2025, viewed 8 November 2026. https://arxiv.org/abs/2411.15114

16. HCAST: Human-Calibrated Autonomy Software Tasks. David Rein, Joel Becker, Amy Deng and others, METR, arXiv:2503.17354, 21 March 2025, viewed 8 November 2026. https://arxiv.org/abs/2503.17354

17. AlpacaEval: An Automatic Evaluator for Instruction-following Language Models. tatsu-lab, GitHub README, viewed 8 November 2026. https://raw.githubusercontent.com/tatsu-lab/alpaca_eval/main/README.md

18. 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

19. Pricing. Claude Platform documentation, Anthropic, viewed 13 November 2026. https://platform.claude.com/docs/en/about-claude/pricing

20. Batch processing. Claude Platform documentation, Anthropic, viewed 8 November 2026. https://platform.claude.com/docs/en/build-with-claude/batch-processing

21. Batch API. OpenAI API documentation, OpenAI, viewed 8 November 2026. https://developers.openai.com/api/docs/guides/batch

22. Batch API. Gemini API documentation, Google, viewed 8 November 2026. https://ai.google.dev/gemini-api/docs/batch-api

23. Batch Mode. Inspect user guide, UK AI Security Institute, viewed 8 November 2026. https://inspect.aisi.org.uk/models-batch.html

24. Caching. Inspect user guide, UK AI Security Institute, viewed 8 November 2026. https://inspect.aisi.org.uk/caching.html

25. Setting Limits. Inspect user guide, UK AI Security Institute, viewed 8 November 2026. https://inspect.aisi.org.uk/setting-limits.html

26. Tasks. Inspect user guide, UK AI Security Institute, viewed 8 November 2026. https://inspect.aisi.org.uk/tasks.html

27. Berkeley Function Calling Leaderboard (BFCL) V4. Gorilla project, UC Berkeley, last updated 12 April 2026, viewed 8 November 2026. https://gorilla.cs.berkeley.edu/leaderboard.html

28. BFCL V4 overall results (data_overall.csv). Gorilla project, UC Berkeley, last updated 12 April 2026, viewed 8 November 2026. https://gorilla.cs.berkeley.edu/data_overall.csv

29. SWE-bench Leaderboards. SWE-bench team, viewed 8 November 2026. https://www.swebench.com/

30. SWE-bench Lite. Carlos E. Jimenez, John Yang and Jiayi Geng, SWE-bench, viewed 8 November 2026. https://www.swebench.com/lite.html

31. swebench.yaml, mini-SWE-agent repository, main branch. SWE-agent project, viewed 8 November 2026. https://raw.githubusercontent.com/SWE-agent/mini-swe-agent/main/src/minisweagent/config/benchmarks/swebench.yaml

32. 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

33. Evidence for inference scaling in AI cyber tasks: Increased evaluation budgets reveal higher success rates. AI Security Institute, 5 March 2026, viewed 8 November 2026. https://www.aisi.gov.uk/blog/evidence-for-inference-scaling-in-ai-cyber-tasks-increased-evaluation-budgets-reveal-higher-success-rates

34. LLM inference prices have fallen rapidly but unequally across tasks. Ben Cottier and others, Epoch AI, 12 March 2025, viewed 8 November 2026. https://epoch.ai/data-insights/llm-inference-price-trends

35. SWE-bench README. SWE-bench project, GitHub, viewed 8 November 2026. https://raw.githubusercontent.com/SWE-bench/SWE-bench/main/README.md

About this report

Published by Rillor on 14 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. What it costs to evaluate an agent. 14 November 2026. https://rillor.com/insights/what-agent-evaluation-costs

Notices

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

Work with Rillor

Send us the tasks your agent handles, the model versions it runs on and the decisions it is trusted with. Contact us at rillor.com/contact?type=evaluation or contact@rillor.com. More on SI evaluation: rillor.com/si/evaluation