

What Counts as a Control? A Scaffold Effect Is a Contrast Conditional on What Was Held Fixed, the 2026 Controlled Studies Each Held Something Different, and Only a Declared Spend Interval Makes a Harness Claim Refutable

Pranay Mahendrakar

SONYTECH · ORCID 0009-0003-7224-029X

pranaymahendrakar@sonytech.in · mahendrakarpranay@gmail.com · +91 6361723454

Abstract. A language-model agent is a model wrapped in a harness that builds its context, exposes its tools, decides when to retry and when to stop. Several 2026 studies set out to measure how much the harness matters relative to the model, holding the model and the tasks fixed, and their answers do not stack: within-model gaps of 0 to 8 points on one terminal benchmark alongside a 40-fold spread in tokens per solved task, 8.5 to 13 points on a coding subset, and up to 28 points on a question-answering benchmark. In this analysis the author reviews and synthesises published results to show that these are not noisy estimates of one quantity. A harness effect is a contrast between two systems conditional on a control set - what was held fixed and what was left free - and a readout, and each study fixed something different: native budget controls, a shared step cap, an attempt count, or nothing at all. The paper argues four points. A cap is not a spend, so a matched ceiling leaves consumption free and lets a harness win by spending. Accuracy is purchasable with attempts, time and retries, so a harness claim with no declared budget excludes no observation. Model and harness interact, so a harness main effect is a property of the model grid. And run-to-run variance is of the same order as many reported gaps. The paper consolidates the measured values in a table and a figure, separates three questions that call for three different controls, states a decision procedure as an algorithm, and names the studies that would settle what remains open.

Keywords: agent evaluation; agent harness; scaffolding; compute-matched baselines; cost-controlled evaluation; experimental design; benchmark validity; LLM agents

SONYTECH ORCID: 0009-0003-7224-029X

1. Introduction

The phrase "the same model" has stopped identifying an agent. Between the language model and the task sits a layer of software that decides what the model sees at each step, which tools it may call and in what format, how errors are reported back, how the growing history is compressed, whether a candidate answer is checked before it is submitted, and when the run ends. Different papers call this layer a scaffold, an agent framework or, increasingly, a harness (Guo et al., 2026). Its influence on measured performance has been visible since the first software-engineering agents: SWE-agent's authors reported that a purpose-built interface for editing and navigating code substantially improved what the same model could do (Yang et al., 2024), and a three-stage pipeline with no autonomous tool use, Agentless, matched or beat the open-source agents of its day on SWE-bench Lite at a reported cost of 0.70 dollars (Xia et al., 2024).

By 2026 the observation had hardened into a thesis. A position paper argues that among frontier models of comparable capability, harness configuration explains more of the variance in long-horizon agent performance than model choice does, and that leaderboards therefore credit models with gains their harnesses produced (Zhang et al., 2026a). Another argues that coding benchmarks collapse model, harness and environment into one score, any part of which can shift the result by

as much as a model generation (Gorinova et al., 2026). A survey organises the whole agent literature around the question of whether the bottleneck sits in the model, the harness, or their coupling, and lists value-aware evaluation among the challenges it leaves open (Guo et al., 2026).

Three controlled studies then tried to measure the harness effect directly, and the trouble starts there. Vats and Golev (2026) held two models fixed across three open-source coding harnesses on fifty terminal tasks and found within-model pass-rate differences of at most eight points, with confidence intervals that include zero, the largest only at its boundary, together with a forty-fold difference in tokens per solved task. Zhang et al. (2026a) ran three models under three harness configurations of their own design on a 100-task subset of SWE-bench Verified and found that changing the harness moved one model by 13 points and the other two by 8.5, while changing the model inside a fixed harness moved scores by 2.5 to 5 points. Starace (2026) ran five models under three scaffolds on the GAIA validation set in a pre-registered design and found scaffold choice moving accuracy within one model by as much as 28 points. An observational audit of 254 SWE-bench submissions found within-model scaffold ranges reaching 29.8 points, against an 8.8-point spread among the top thirty entries (Liu et al., 2026).

Read side by side, these numbers invite a meta-analytic question: what is the size of the harness effect, and on which side of the model effect does it fall? In this

analysis the author reviews and synthesises published results to argue that the question has no answer as posed, and that the reason is not sampling error. Each study held a different set of things fixed. Vats and Golev deliberately left each harness's own budget controls in place, on the grounds that those controls are part of what a deployer buys. Zhang et al. imposed a shared 50-step cap and let token and dollar spend vary. Starace fixed tasks and attempts, disabled reasoning for every model, and did not match spend at all. Liu et al. had no control, since submissions were built by different teams under different budgets. A harness effect is a contrast between two systems, and a contrast is defined only relative to what is held constant. When the constant changes, the contrast is a different quantity.

That observation would be a methodological footnote if the free variable were inert. It is not. Agent accuracy is purchasable: repeated sampling raised the fraction of SWE-bench Lite issues solved by one open model from 15.9 to 56 percent (Brown et al., 2024), eight attempts doubled the best MLE-bench score (Chan et al., 2024), and the comparison between AI agents and human experts on research-engineering tasks reverses as the time budget grows (Wijk et al., 2024). A harness is, among other things, a policy for converting budget into attempts, checks and retries. If the budget is not fixed, a harness can win by spending; if the budget is fixed only as a ceiling, a harness can win by spending more of the ceiling. Kapoor et al. (2024) made the general point for agents two years ago: evaluations that do not control cost reward agents that are needlessly expensive and mislead the field about where accuracy gains come from. The 2026 harness literature reintroduces the confound at a new layer.

The paper's question, then, is the one in its title. What counts as a control for a claim that one harness is better than another? Section 2 fixes terms and separates three questions a harness comparison can answer: which system to deploy, what a design change is responsible for, and how much capability a model holds. Section 3 reads the controlled studies against one another and shows what each held fixed. Section 4 argues that caps are not spends. Section 5 assembles the evidence that agent accuracy is purchasable and draws the consequence for refutability. Section 6 shows that model and harness interact strongly enough that a harness main effect is a property of the grid. Section 7 sets the reported gaps against the measured noise floor. Section 8 argues that the scoring rule and the exclusion rule belong to the control set. Section 9 adds the design-time budget, where the older machine-learning literature on hyperparameter tuning already learned the lesson this one is relearning. Section 10 consolidates the numbers in Table 1 and Figure 1. Section 11 states which control answers which question, in Table 2 and Algorithm 1, and the condition under which a harness claim is refutable. Section 12 positions the argument against its precursors, Section 13 names the studies that would settle the open part, Section 14 states what is not known, and Section 15

gives the case against this paper's own argument.

2. Terms and scope

Harness. Following Guo et al. (2026) and the disclosure layers of Zhang et al. (2026a), the harness is everything in an agent that is not the model's weights: the context-construction policy, the tool set and its schemas, the error contract, the control loop with its retry and escalation rules, any memory or state store, and any verification step. The terms scaffold and harness are used interchangeably here, as they are in the studies under review. A multi-agent architecture is one kind of harness; so is a single model running a planner-then-executor loop.

Contrast. A harness claim compares two systems, ($m, h1$) and ($m, h2$), that share a model m and differ in harness. The claim is about a difference in some readout R , averaged over a task distribution T . Written out, a harness effect is

$$\text{delta}(h1, h2 \mid m, T, C, R)$$

where C is the control set: every factor other than the harness that was held fixed, together with the rule by which each free factor was allowed to vary. This notation is a definition, not a model of anything; its only job is to make visible that delta takes a different value for every different C .

Control set. Seven components recur in the studies reviewed, and each is a place where two studies can differ. (1) Model settings: the model identifier and its inference settings, notably reasoning effort, which providers define in incompatible ways (Kapoor et al., 2025). (2) The task set and the slice rule: which items are scored, and which runs are excluded as infrastructure failures. (3) The run-time budget: a cap on steps, turns, tokens, wall-clock time or dollars, and whether the cap or the consumption is what is matched. (4) Attempts: how many independent tries each task receives and how they are aggregated. (5) The design-time budget: how much search, tuning or adaptation went into the harness, and on which data. (6) The readout: pass rate, pass^k, tokens or dollars per solved task, or a frontier. (7) Replication: how many runs per cell and how variance is reported.

Three questions. A harness comparison can serve three different purposes, and the argument of Section 11 is that each needs a different control. The deployment question asks which (model, harness) pair a team should run, given a price and a latency tolerance. The attribution question asks whether a design change - a verifier, a context policy, a planning step - is responsible for a gain, as opposed to the extra compute the change consumes. The elicitation question asks how much capability a model holds when given the best available harness, a question safety evaluations and capability forecasts depend on (Davidson et al., 2023; Kwa et al., 2025). Kapoor et al. (2024) separated the needs of model developers from those of downstream developers; the

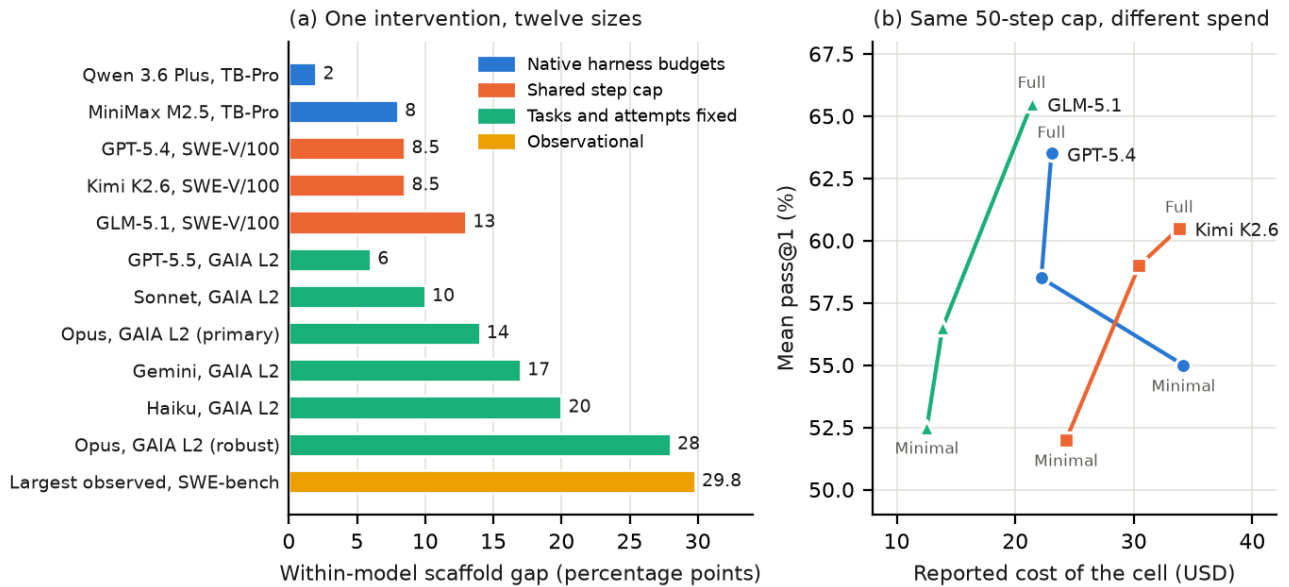


Figure 1. (a) Largest within-model accuracy difference between scaffolds for one model in one study cell, in percentage points, coloured by what the study held fixed: native harness budget controls (Vats and Golev, 2026, Table 2, Terminal-Bench Pro, 50 tasks), a shared 50-step cap (Zhang et al., 2026a, Table 2, SWE-bench Verified subset of 100), tasks and three attempts with spend free (Starace, 2026, Section 4.1.3, GAIA Level 2, primary slice for all five models and, for Opus, the robust slice that excludes provider-defect units), and an observational leaderboard maximum (Liu et al., 2026). (b) Mean pass@1 against reported dollar cost for the nine cells of Zhang et al. (2026a), Tables 2 and 9; lines join the minimal, improved and full harness for each model. Under the same step cap the full harness cost more than the minimal one for GLM-5.1 and Kimi K2.6 and less for GPT-5.4, in dollars; token counts move the other way.

Table 2. Which control set answers which harness question.

Question	Who asks it	Budget control	Readout	Model grid
Deployment	A team choosing a system	Native, as shipped; report spend	Cost per solve, latency, pass ^k	The models to be deployed
Attribution	A method paper	Matched consumption, over an interval	Accuracy-spend frontier	Two or more families
Elicitation	Safety or forecasting	Unbounded or very large; report the curve	Best attainable, with curve	The model under assessment

three-way split here refines that distinction by adding attribution, which is the question most method papers implicitly ask.

Scope. This paper concerns agents built on language models and evaluated on task benchmarks with automatic or rubric scoring. It does not re-argue the case on multi-agent debate and persona schemes against self-consistency at matched compute, which has its own literature (Zhang et al., 2025; Tran and Kiela, 2026; Xu et al., 2026); multi-agent architectures appear here only as one kind of harness and as the setting in which the cap-versus-spend problem was first documented in detail. Nor does it argue for or against any particular harness. The paper’s claims are about what an evaluation can and cannot license.

3. The same intervention at four sizes

3.1 Native controls: Vats and Golev

Vats and Golev (2026) evaluated two models, Qwen 3.6 Plus and MiniMax M2.5, under three open-source harnesses, Goose, OpenCode and the OpenHands SDK, on a stratified subset of fifty Terminal-Bench Pro tasks, for 300 trials in all. Per-cell pass rates ranged from 48 to 50 percent for Qwen and from 38 to 46 percent for

MiniMax, and every per-cell 95 percent interval spanned roughly plus or minus fourteen points. Within a model, the harness range was 2 to 8 points; across models within a harness, 4 to 10 points. On pass rate, then, this study finds harness and model effects of comparable and small size, with most pairwise differences statistically indistinguishable from zero.

On tokens the picture is the reverse. The ordering of tokens per solved task was the same for both models: Goose cheapest, then OpenHands SDK, then OpenCode, which consumed about forty times as many tokens per solved task as Goose. The bootstrap intervals do not overlap: for Qwen, Goose’s interval is 21 to 40 thousand tokens and OpenCode’s 733 thousand to 1.84 million. The authors report that the gap is not explained by turn counts, which were similar, and attribute it, with appropriate hedging, to per-turn context growth, tool serialisation and harness-specific accounting.

The design decision that matters for this paper is stated explicitly. Goose and the OpenHands SDK were given a 40-turn maximum; OpenCode exposes no turn-budget flag through the evaluation framework and was bounded only by the 900-second wall-time cap that applied to every trial. The authors chose to evaluate each harness under its native control surface, treating its budget

controls as part of the system under study rather than as nuisance parameters to equalise. That is a coherent choice for the deployment question: a team that installs OpenCode gets OpenCode's budget behaviour. It is the wrong choice for the attribution question, and the authors do not claim otherwise.

3.2 A shared step cap: Zhang et al.

Zhang et al. (2026a) built three harness configurations - a minimal open-loop baseline that appends every prior step to the context, an improved configuration with compressed, relevance-retrieved context, a smaller tool schema and structured retry on tool failure, and a full configuration that adds per-step self-checking, drift checks, loop detection, output validation and checkpoint rollback - and ran each under three models on a difficulty-stratified 100-task subset of SWE-bench Verified, two runs per cell. All cells shared a 50-step budget and a 120-second per-step timeout. Mean pass@1 rose from minimal to full by 13 points for GLM-5.1 (52.5 to 65.5), 8.5 for GPT-5.4 (55.0 to 63.5) and 8.5 for Kimi K2.6 (52.0 to 60.5). Within any one harness, the three models differed by 3.0, 2.5 and 5.0 points. The authors report harness-induced variances of 12.17 to 29.56 squared points against model-induced variances of 1.17 to 4.22, an aggregate ratio of 7.80, and six model-pair ranking reversals out of nine possible comparisons.

The step cap is a real control: no harness could run longer than fifty steps. It is not a control on spend. The paper's resource appendix reports the dollar cost and token count of each cell. For GLM-5.1, the full harness cost 21.39 dollars against 12.54 for the minimal one, while delivering the 13-point gain. For Kimi K2.6, the full harness cost 33.83 dollars against 24.29. For GPT-5.4 the order reverses: the minimal harness was the most expensive cell at 34.19 dollars, and the improved and full harnesses cost 22.19 and 23.05. Figure 1(b) plots these nine published dollar points. The token column points the other way: the full harness consumed fewer tokens than the minimal one for every model - 24.5 against 29.7 million for GLM-5.1, 47.7 against 60.4 million for Kimi K2.6, and 10.0 against 18.9 million for GPT-5.4. So in two of the three models the harness that scored best also cost the most in dollars, while in tokens it used less than the minimal harness in all three; whether the gain coincided with more spend depends on the unit in which spend is counted. The step cap settled neither question, because a cap on the number of steps says nothing about what each step costs. The authors' own remarks on cost compare the full harness with the improved one, not with the minimal one: in aggregate they find the improved harness "the cheapest controller in this implementation", with the full harness spending "more on verification and recovery to obtain the best pass rate", and they note that the full harness "frequently spends more steps and tokens" than the improved one. For GPT-5.4, the minimal harness's append-everything context made each step expensive and the better harnesses were cheaper in both units as well as more accurate. The study therefore contains, in one grid, cases

where the harness effect coincides with more dollar spend, a case where it coincides with less spend in both units, and two cases where dollars and tokens disagree on which harness spent more. It cannot say how much of the GLM-5.1 gain the design would retain at the minimal harness's dollar spend, because no cell was run at matched cost.

There is also a second difference between the minimal and the other configurations that the step cap does not touch: the minimal harness is given a context cap of roughly 200 thousand tokens and no compression, the other two a cap of roughly 32 thousand with summarisation and retrieval. The minimal harness also has a verbose tool schema and no retry logic, so context policy, tool schema, retry and verification are varied together. This is a fair design for showing that a harness bundle can dominate a model choice, which is what the paper sets out to show, but it is not a design that isolates any one component.

3.3 Tasks and attempts fixed: Starace

Starace (2026) ran a pre-registered three-by-five factorial: a ReAct loop, a planner-actor-rater multi-agent design and a planner-then-executor design, under three Anthropic models and one model each from Google and OpenAI, on the 53 Level 1 and 86 Level 2 questions of the GAIA validation set (Mialon et al., 2023), three attempts per question. Tasks, attempts and conditions were held fixed. Spend was not: the study records realised dollar cost per correct answer for every cell as an outcome, not as a constraint. Reasoning was pre-registered as enabled at default budget and was in fact disabled for every model, because pilot runs showed reasoning cost scaling unpredictably across providers; the author reports the deviation and notes that it is constant across cells.

The headline 28-point gap is the Level 2 range for the most capable Anthropic model under what the study calls the robust slice, which drops attempt-units flagged with a provider serialisation defect. Under the primary slice, which keeps them, the same gap is 14 points. The Level 2 primary-slice gaps for the five models are 20 points for Haiku, 10 for Sonnet, 14 for Opus, 17 for Gemini and 6 for GPT-5.5. The defect selectively suppressed the structured scaffolds' accuracy for the Anthropic models, so removing it widens the gaps rather than narrowing them. The size of the harness effect for one model at one level therefore depends on an exclusion rule, which Section 8 takes up. The study's own conclusion is careful: single-scaffold capability numbers are conditional on the scaffold, and the elicitation gap need not shrink as models improve.

3.4 No control: the leaderboard

Liu et al. (2026) audited 254 SWE-bench submissions across four splits without running anything. Within-model scaffold ranges reached 29.8 points, more than three times the 8.8-point spread among the top thirty Verified entries. Interaction tests between model and scaffold were significant in six of nine cases after

correction. The authors are explicit that the design is observational and does not identify causal scaffold effects. That caution is the point: a leaderboard row is a (model, harness, budget, design effort, team) bundle, and its within-model range mixes harness design with every other difference between submitters.

3.5 What the four have in common

Four studies, four control sets, and within-model harness gaps of 2 to 8, 8.5 to 13, 6 to 28, and up to 29.8 points. Figure 1(a) arranges them by the kind of control each used. The tempting reading is that the true harness effect lies somewhere in that range and the studies are noisy draws. The reading this paper argues for is that the studies estimate different contrasts: Vats and Golev estimate the difference between deployable products under their own budget behaviour, on a benchmark where the models are near half accuracy; Zhang et al. estimate the difference between three designs of increasing closed-loop control under a shared step ceiling and free spend; Starace estimates the difference between three control-loop topologies with reasoning disabled and spend free, conditional on an exclusion rule; Liu et al. estimate nothing causal. A pooled number would average contrasts that do not share a definition.

4. A cap is not a spend

The most common move toward fairness in the reviewed studies is to impose a shared ceiling - on steps, turns, iterations or requested reasoning tokens. A ceiling bounds consumption from above. It does not equalise consumption, and the harness is the component that decides how much of the ceiling is used.

Kim et al. (2025) show how hard it can be to tell from a paper whether a ceiling or a consumption was matched. Their study of 260 configurations across six agentic benchmarks, five architectures and three model families matched the maximum total iterations between single-agent and multi-agent systems "to ensure computational fairness", and it states that "All MAS and SAS configurations were matched for total reasoning-token budget (mean 4,800 tokens per trial)". Yet the same paper reports overheads for the multi-agent architectures relative to the single agent of 58 percent for independent agents, 285 percent for centralised, 263 percent for decentralised and 515 percent for hybrid. Its methods section defines these as counts of "inter-agent message exchanges required per task", while a later section describes them as representing 1.6 to 6.2 times the single agent's token budget "at matched performance". The paper does not reconcile the two readings. If the second holds, the matched iteration cap left consumption free by up to a factor of six; if the first holds, tokens were matched and the overhead is in messages. The paper's range of outcomes - from an 80.8 percent relative gain for a centralised architecture on a decomposable financial-reasoning task to a 70.0 percent loss for independent agents on sequential planning - is a range across tasks and architectures, not across controls, and it

is properly read as evidence that architecture-task fit matters. What the cap-versus-spend question adds is that, under the token reading of the overhead, the gains on the favourable tasks were bought at a token cost the matched cap does not express, and a reader of the paper cannot tell which reading applies.

Tran and Kiela (2026) document the mechanism one level down. Comparing single-agent and multi-agent systems at matched requested thinking budgets, they found that for Gemini 2.5 Flash the visible reasoning text of the single-agent pipeline plateaus: about 354 tokens at a requested budget of one thousand and 359 at ten thousand. At a matched requested budget of one thousand on Gemini 2.5 Pro, the sequential multi-agent pipeline produced 693 tokens of visible reasoning against 390 for the single agent, because it made several calls, each of which used part of its own allowance. The provider documentation treats the budget as guidance rather than a hard limit, and the authors found the API-reported token counts unreliable. Their resolution is candid: the requested budget is the only quantity the experimenter can set, so they matched it, knowing that consumption was not matched. HAL found the same problem across providers: one widely used library maps "high" reasoning effort to 4,096 tokens while a provider does not disclose what its own settings mean, so that equal nominal effort cannot be guaranteed to mean equal compute (Kapoor et al., 2025).

The harness studies show the same structure in different units. Zhang et al. (2026a) fixed fifty steps and observed a 71 percent dollar-cost increase from minimal to full harness for one model and a 33 percent decrease for another. Vats and Golev (2026) gave two harnesses a 40-turn cap and the third none, and observed similar turn counts with a forty-fold token spread, because the cap was on a unit the harnesses did not differ in. Shen et al. (2025) show that the interaction horizon - the number of environment steps an agent takes within one rollout - is itself a scaling axis on which web-agent success improves, which means a step cap is a compute cap in one unit and silent about the others.

The consequence is not that caps are useless. A cap is the right control when the claim is about behaviour under a deployment limit: a harness that solves more tasks within fifty steps is a better harness for a team that will stop it at fifty steps. The consequence is that a cap cannot support an attribution claim. If harness h2 beats h1 under a shared cap and also consumes more of it, the design and the spend are confounded, and nothing in the comparison separates them. The 2024 lesson was that accuracy-only comparisons reward spending (Kapoor et al., 2024); the 2026 refinement is that capped comparisons reward spending within the cap.

5. Accuracy is purchasable, and what follows for refutation

5.1 The purchase curve

That a harness can win by spending matters only if spending buys accuracy at the margins the harness literature reports. The evidence that it does is broad.

Attempts. Repeated sampling on SWE-bench Lite raised the fraction of issues with at least one correct sample from 15.9 percent at one sample to 56 percent at 250, above the single-attempt state of the art of the time (Brown et al., 2024). On MLE-bench, o1-preview's medal rate doubled from 16.9 percent at one attempt to 34.1 percent at eight, and GPT-4o's rose from 8.7 to 11.8 percent when the time allowance per competition went from 24 to 100 hours (Chan et al., 2024). On GAIA, an enterprise agent reported 75.2 percent pass@1 and 83.0 percent best-of-k (Dastidar, 2026).

Time. RE-Bench compared frontier agents with human experts on seven research-engineering environments: at a total budget of two hours per environment the best agents scored four times higher than the experts, while at 32 hours the experts scored twice as high as the best agent (Wijk et al., 2024). The ordering of two systems reversed with the budget, which is the strongest available illustration that "which is better" is a question about a point on a curve.

Retries with a cheap verifier. Kapoor et al. (2024) re-evaluated three HumanEval agents from the leaderboard of the time - LDB, LATS and Reflexion (Shinn et al., 2023; Zhou et al., 2024) - against simple baselines that retry a model up to five times when the provided tests fail, with or without raising the temperature on each retry. They found no significant accuracy difference between the temperature-raising retry and the best agent, while Reflexion and LDB cost more than half again as much and LATS more than fifty times as much.

Inference-time compute in general. Snell et al. (2024) show that allocating test-time computation adaptively can be more effective than scaling parameters in some regimes, and Davidson et al. (2023) estimate that most of the post-training enhancements they survey - a set that includes scaffolding - are worth more than a fivefold increase in training compute, and some more than twenty-fold. Chen et al. (2024) add the caveat that more calls are not always better: majority-vote performance can rise and then fall with the number of calls, because extra samples help on easy queries and hurt on hard ones. The purchase curve is real but not monotone, which only strengthens the case for reporting it rather than a point.

5.2 Simpler harnesses at lower spend

The second strand is that when spend is examined, much of the reported advantage of elaborate harnesses shrinks. Agentless matched the open-source agents of its time at low cost with a fixed three-phase pipeline (Xia et al., 2024). Jiang et al. (2025) put a whole repository into the context of a long-context model with no tools and

reported 38 percent on SWE-bench Verified for Gemini 1.5 Pro, against 32 percent for a carefully tuned scaffold with the same model, and 50.8 percent for Gemini 2.5 Pro by the same unscaffolded method. Lindenbauer et al. (2025) compared context-management policies inside SWE-agent across five model configurations and found that simply masking old observations halved cost relative to the raw agent while matching the solve rate of LLM summarisation. Wang et al. (2025) reported a GAIA agent that retained 96.7 percent of a leading open-source framework's performance while cutting its operational cost from 0.398 to 0.228 dollars. Moghe and Chin (2026) lifted a 15.5 percent one-shot ARC-AGI-1 baseline to 57.5 and then 67.25 percent pass@2 with two harness designs, at 0.25 and 0.62 dollars per task - a harness gain reported together with its price, which is the form this paper argues for.

None of these results shows that elaborate harnesses are worthless. In Starace (2026) a single cell, Gemini with the planner-then-executor scaffold, had the lowest cost per correct answer on both GAIA levels and the highest accuracy of any cell on Level 2, and in Zhang et al. (2026a) the two better harnesses for GPT-5.4 were also cheaper than the minimal one. The point is narrower. Without the spend axis, a reader cannot tell whether a harness gain is a design gain or a purchase, and the published record contains both.

5.3 Why an undeclared budget makes a claim unrefutable

A claim is refutable when some possible observation would contradict it. Consider the claim "harness h2 is better than h1 for model m on tasks T". If the claim does not say at what spend, then any observed deficit of h2 can be explained as insufficient budget for h2's more elaborate loop, and any observed advantage can be credited to the design, whatever h2 consumed. Kapoor et al.'s observation that maximising accuracy alone can drive cost without bound is the same point from the other side: if accuracy is the only readout and spend is free, the comparison has a winner by construction - whichever system spent more on the margin where spend still buys accuracy.

The claim becomes refutable once it names a spend interval and a dominance relation: "over spend s in $[a, b]$ per task, measured in dollars at stated prices and in tokens, the accuracy of $(m, h2)$ at spend s exceeds the accuracy of $(m, h1)$ at spend s ". One configuration of h1 that sits above h2 anywhere in the interval refutes it. That is a frontier claim, and it is the form Dodge et al. (2019) argued for when they showed that several published model comparisons in natural language processing would have reached a different conclusion with more or less tuning computation. It is also the form Kapoor et al. (2024) recommended for agents as a Pareto frontier of accuracy against cost. What the 2026 harness studies add is that the spend axis needs to be consumption rather than a ceiling, for the reasons in Section 4, and that it needs to be stated per question, for the reasons in Section 11.

6. There is no harness main effect to estimate

A harness effect averaged over models is meaningful when the ranking of harnesses is the same for every model. The evidence is that it often is not.

Zhang et al. (2026a) report six ranking reversals among nine model-pair and harness-pair comparisons, and treat them as evidence that the interaction term is substantial. Starace (2026) pre-registered the prediction that more capable models would be less sensitive to scaffold choice; the prediction was rejected in direction at Level 2, where the most capable Anthropic model gained most from structured scaffolds, and the multi-agent scaffold's advantage at Level 2 appeared within the Anthropic family but not for the Google or OpenAI models. The conditioning variable was model family, not capability. Liu et al. (2026) found significant model-by-scaffold interactions in six of nine observational tests.

The interaction runs through training as well as inference. Thangarajah et al. (2026) report that open models fine-tuned on trajectories collected under one harness, OpenHands, score well under it and degrade substantially under any other, while untrained base models show no such divergence; the coupling is induced by fine-tuning. HarnessDev found that the gains from a harness evolved by a model depend strongly on which model then executes it, with limited transfer across models (Wu et al., 2026). Kimi-Dev and Clean-PR move capability in the other direction, training structure that a harness would otherwise supply into the weights (Yang et al., 2025; Zhu et al., 2026). When the boundary between model and harness can be moved by training, "the model held fixed" is itself a design choice.

Interactions also appear inside harness components. Zhang et al. (2026c) swept observation masking across agent backbones from 4 to 284 billion parameters and found the accuracy gain from masking traced an inverted U against the model's accuracy without context management - flat with weak retrievers, peaking for mid-capacity models with strong retrievers, collapsing for saturated models. Wiedmann et al. (2026) varied five configuration axes of a coding agent on scientific tasks and found that extra time helped only when the agent had enough task information or a capable enough model. Weinberger and Hozes (2026) found that an intervention which saves cost under one coding harness need not save it under another, and treat prompt wording, inference effort and harness as factors that interact rather than as settings that can be fixed one at a time. SWE-Effi found that an AI system's effectiveness under resource limits depended on how well scaffold and base model fit together, not on the scaffold alone (Fan et al., 2025).

Two consequences follow. First, a harness main effect averaged over a grid of models is a property of the grid. Zhang et al. (2026a) do not claim that their 7.80-fold ratio is universal, and they caution that the interaction statistic they propose alongside it, partial eta-squared, is a fixed-effects quantity on the chosen grid, positively biased in small grids, and should be reported with an

interval. The same reasoning applies to the variance ratio, which is computed over the same three-by-three grid. The same holds for the claim that harness variance exceeds model variance. That claim is stated for frontier models of comparable capability, and model-induced variance is small in that regime partly by construction: models chosen because they score alike will produce a small spread. Second, the harness that is best for one model is not evidence about another. A study that varies harnesses under one model licenses a claim about that model only.

7. The noise floor

The reported harness gaps need to be read against how much an agent's score moves when nothing is changed.

Wiedmann et al. (2026) collected more than 18,000 trajectories of a coding agent asked to find and operate published specialist models on four scientific tasks, and decomposed the variance: among genuine attempts, about 54 percent of the spread in scores was attributable to re-running an unchanged configuration, and less than half to the configuration changes themselves. Sahu and Pandey (2026) decomposed the variance of a repaired computer-use policy and found the run-to-run distribution on the hardest environment bimodal, with the data draw carrying the largest share; they estimate that a single-run improvement of the size typically published would have the wrong sign about a third of the time in a comparable regime. Bhat et al. (2026) report that 23 repeated evaluations of the same setup on one tool-use benchmark produced scores from 57.9 to 76.8 percent. Kozyrev et al. (2026) found that a 4.9-point gain from an optimised repository document could not be separated from the agent's run-to-run variance at the number of tasks one repository supplies.

Sampling of tasks adds to sampling of runs. A pass rate near one half on 100 independent items has a binomial standard error of about five points; on fifty items, about seven. That arithmetic, which Miller (2024) sets out with the formulas for paired comparisons, is why Vats and Golev's per-cell intervals span about plus or minus fourteen points, and why Liu et al. (2026), applying exact paired tests, could separate none of the 29 adjacent pairs among the top thirty SWE-bench Verified entries, though the larger test split separated 14 of 23. Against this floor, Zhang et al.'s within-harness model differences of 2.5 to 5 points on 100 tasks and two runs are of the size the design cannot resolve, and some of the six ranking reversals involve model pairs separated by about that much. Their harness differences of 8.5 to 13 points are larger and more likely to survive, though the main grid reports cell means over two runs rather than intervals.

The older literature has been here. Henderson et al. (2017) showed that in deep reinforcement learning the variance across random seeds and environment non-determinism could make reported improvements over baselines uninterpretable. Agarwal et al. (2021) showed, on Atari 100k, that conclusions drawn from point

estimates over a handful of runs diverged from those drawn with interval estimates, and proposed reporting intervals and robust aggregates. Bouthillier et al. (2021) modelled the whole benchmarking process and showed that data sampling, initialisation and hyperparameter choice each move results markedly. The agent literature's version adds a term those papers did not have, the stochasticity of a hosted model's API, which a harness cannot remove and which a single run cannot estimate.

The noise floor is not a separate issue from the control set. It is the seventh component of it. A harness effect estimated from one run per cell is a contrast between two draws, and its sign carries limited information when the run-to-run share of variance is as large as the 54 percent Wiedmann et al. report. Replication count and the variance decomposition belong in the declared control set for the same reason the budget does: without them, two studies that report the same point estimate are not making the same claim.

8. The readout and the exclusion rule are part of the control

A harness effect is a difference in a readout, and the readout has its own validity conditions.

The exclusion rule changed Starace's headline gap by a factor of two (Section 3.3). The rule was reasonable and disclosed - attempt-units that hit a provider serialisation defect do not measure the scaffold - but a different study with a different rule, or none, would report a different gap for the same model and scaffolds. Lip et al. (2026) show how much a benchmark's own infrastructure can contribute. In a production record of 639 scored terminal tasks and 28,801 trials, 125 tasks had no honest pass; after a validity screen, only 78 of them survived as certified-unsolved, while 14 had broken oracles, 8 were dominated by infrastructure failures, 4 were passable only through a verifier bypass and 21 could not be certified either way. A harness that handles a timeout or a flaky container differently will score differently on such tasks for reasons that are not about agent capability.

The scorer itself can move the readout. Zhu et al. (2025) found that issues in task setup or reward design can misestimate agent performance by up to 100 percent in relative terms, citing insufficient tests in SWE-bench Verified and a benchmark that counted empty responses as successes. Bhat et al. (2026) found evaluator-human disagreement on 18.5 percent of 496 expert-reviewed tool-calling tasks. Zhang et al. (2026b) name a double measurement confound: execution-critical decisions taken by a fixed scaffold rather than the model, and a scorer that may not track task correctness; they find scaffold ownership to be an uncontrolled axis wherever they probed it. HAL's authors report that one benchmark embedded agent instructions in its task text, entangling task specification with scaffold (Kapoor et al., 2025). Bean et al. (2025), reviewing 445 LLM benchmarks, document the construct-validity problems that follow when the measured quantity is not tied to the

phenomenon of interest.

The aggregation rule is a readout choice too. `pass@1`, `best-of-k`, and `pass^k` - the probability that all k trials succeed, which `tau-bench` introduced to measure consistency (Yao et al., 2024) - can rank the same systems differently, because a harness that converts budget into more independent attempts raises `best-of-k` without changing per-attempt reliability. `Cost-of-pass`, the expected dollar cost of one correct solution, folds accuracy and price into one number (Erol et al., 2025); Vats and Golev's tokens per solved task is a token-denominated version of the same idea. These readouts answer different questions, and a harness effect in one is not a harness effect in another.

9. The design budget

The run-time budget is the one the reviewed studies argue about. The design-time budget - how much search, tuning or adaptation produced the harness - is the one they mostly leave unstated, and it is the one the older literature learned most painfully to report.

Lucic et al. (2017) ran a large neutral study of generative adversarial networks and found that most models could reach similar scores given enough hyperparameter optimisation and random restarts, and no tested algorithm consistently beat the original non-saturating formulation; they read the published improvements as plausibly the product of larger tuning budgets. Melis et al. (2017) re-tuned language-model architectures with large-scale automatic search and found a properly regularised standard LSTM ahead of more recent designs. Musgrave et al. (2020) found that four years of claimed advances in deep metric learning were marginal at best once methodological flaws were removed. Dodge et al. (2019) turned the lesson into a reporting device: expected validation performance of the best configuration found, as a function of the number of tuning trials, so that a comparison is made at matched design budget or shown across a range of them.

Agent harnesses are tuned objects. Automated design of agentic systems uses a meta-agent that writes new agent designs in code, building on an archive of earlier discoveries (Hu et al., 2024). Gideoni et al. (2026) re-examined such code-evolution pipelines and found that, for agentic scaffold design, high variance among candidate scaffolds combined with small evaluation sets led the search to select suboptimal designs, with hand-designed majority-vote scaffolds performing best. Wang et al. (2026) compared three harness optimisers under identical optimisation budgets on Terminal-Bench 2.0 tasks: all three beat the baseline when evaluated once, but when new tasks arrived one optimiser's harness transferred below the unoptimised baseline. StateM reports raising DeepSeek-V4 Flash from 82.7 to 88.1 percent on a terminal benchmark with less than 38 dollars of adaptation, a design-time cost the paper discloses and most do not (Qin et al., 2026). A manually built harness absorbs engineer-weeks of iteration against the very benchmark on which it is then reported; Zhang

et al. (2026a) note the corresponding problem for public harness comparisons, which come from teams with different engineering budgets.

The Arena analysis supplies the leaderboard version. Singh et al. (2025) found that a few providers tested many private variants before release and could choose which score to publish, at an extreme 27 variants from one provider before one release, and showed that the ability to select the best of several biases the published score. A harness iterated against a public benchmark and reported once is a best-of-many selection whose many is unreported. The correction the older literature reached is available: report the number of configurations tried, the data used to choose among them, and, where feasible, performance as a function of design budget.

10. The numbers in one place

Table 1 collects the measured values the argument rests on, one row per cited result. Each row names its source, what the study held fixed, what it left free, and the effect it reported. Figure 1 plots the within-model harness gaps by control type and re-plots the cost and accuracy of the one published grid that reports both for every cell.

Three readings of Table 1 and Figure 1 are warranted and one is not. Warranted: the reported harness gaps span an order of magnitude; the studies that report them held different things fixed; and in the one grid with cell-level costs, dollar cost moved with the harness in two of three models. Not warranted: that the spread in Figure 1(a) is caused by the control type. Four studies on four benchmarks with different models cannot separate the effect of the control from the effect of benchmark, model and harness choice. The figure shows that the numbers are not comparable; it does not show by how much each control moves them.

11. What counts as a control

11.1 Three questions, three controls

The argument so far is negative: each published control answers some question, none answers all of them, and pooling across them averages contrasts with different definitions. The constructive claim is that the right control follows from the question, and that the three questions in Section 2 call for three different ones.

For the deployment question, Vats and Golev's choice is right: the budget behaviour of a harness is part of the product, and equalising it would answer a question the deployer did not ask. What the deployment control must add is the spend readout, in dollars at stated prices and in tokens, and a consistency readout such as pass^k, because a deployer pays for every run, including the failed ones (Fan et al., 2025; Erol et al., 2025). The deployment answer is local to the models and prices of the day and should be read that way.

For the attribution question, the control must match consumption, not ceilings, and must do so across an interval, because a design that helps at low spend may

hurt at high spend and the reverse. The readout is a frontier: accuracy as a function of realised spend for each harness, with the claim stated as dominance over a declared interval. The model grid must include at least two model families, because the interaction evidence of Section 6 shows that a design gain under one family is not evidence about another. Zhang et al.'s factorial protocol is a necessary part of this design; matched spend is the part it lacks.

For the elicitation question, the budget should be as large as the assessment can afford and the result reported as a curve, because the question is what the model can do with the best available support, and the RE-Bench result shows that the answer at two hours and at thirty-two hours can point in opposite directions (Wijk et al., 2024). Starace's conclusion that single-scaffold capability numbers are scaffold-conditional estimates is the elicitation version of this paper's argument, and Davidson et al.'s compute-equivalent gains are its quantitative motivation: if post-training enhancements of this kind can be worth a several-fold increase in training compute, an elicitation that uses a weak scaffold may understate capability by a comparable margin.

11.2 A decision procedure

Algorithm 1 states the procedure the paper proposes for deciding whether a harness claim is licensed. It is a reporting and inference rule, not an experiment; every step can be applied to published work.

The thresholds in steps 4 and 6 - three spend levels, two families, twice the standard error - are proposals, not results; no study has calibrated them. Their purpose is to make the claim's scope explicit enough that a reader can tell which of the three questions it answers and what observation would contradict it.

11.3 Reading the published studies through the procedure

Applied to the studies of Sections 3 and 4, the procedure licenses something in each case, and less than the headline in most. Vats and Golev (2026) license a deployment claim for two models: under their native controls the three harnesses are statistically close on pass rate and an order of magnitude apart on tokens per solved task, so the cheapest harness is on the cost-accuracy frontier for both models, at a pass-rate cost of up to 8 points for MiniMax. Zhang et al. (2026a) license the claim that, for three models of similar capability under a shared step cap, a closed-loop harness bundle raised pass rate by more than the spread between the models - an existence proof that harness variance can exceed model variance in that regime, which is what the authors say they intend. They do not license an attribution of the gain to closed-loop control as opposed to spend, because spend was not matched and dollar cost moved with the harness in two of three models. Starace (2026) licenses an elicitation claim: with reasoning disabled and spend free, the scaffold-conditional capability of each model differs by 6 to 20 points at Level 2 on the primary slice, and the ordering of

Table 1. Harness and budget effects as reported, with the control set each study used.

Source	Setting	Held fixed	Left free	Reported effect
Vats and Golev 2026	2 models, 3 harnesses, 50 TB-Pro tasks	Model, tasks, task instruction, near-identical system prompt, 900 s wall-time cap per trial	Turn cap (40 for two harnesses, none for OpenCode), spend	Pass-rate gap 0-8 pts; 40x tokens per solve
Zhang et al. 2026a	3 models, 3 harnesses, 100 SWE-V tasks, 2 runs	50-step cap, tasks	Context policy, spend	Harness 8.5-13 pts; model 2.5-5 pts
Zhang et al. 2026a	Same grid, resource table	50-step cap	Spend	GLM-5.1 full 21.39 vs minimal 12.54 USD; 24.5M vs 29.7M tokens
Starace 2026	5 models, 3 scaffolds, GAIA L2 (86 Qs)	Tasks, 3 attempts, reasoning off	Spend	Gap 6-20 pts (primary); Opus 14 vs 28 by slice
Liu et al. 2026	254 SWE-bench submissions	Nothing	Everything	Within-model range up to 29.8 pts
Kim et al. 2025	260 configs, 6 benchmarks	Max iterations; reasoning-token budget (as stated)	Messages exchanged; tokens consumed (disputed within source)	Overhead 58-515 pct (messages; also stated as 1.6-6.2x tokens)
Tran and Kiela 2026	Gemini 2.5, multi-hop QA	Requested budget	Consumed tokens	Flash 354 vs 359 tokens at 1k vs 10k
Kapoor et al. 2024	HumanEval agents vs retry	Model	Spend	No sig. accuracy gap; LATS over 50x cost
Kapoor et al. 2025	9 models, 9 benchmarks	Scaffold, benchmark	Reasoning effort	21 of 36: higher effort no better
Brown et al. 2024	SWE-bench Lite, one model	Model, harness	Samples	15.9 pct at 1 to 56 pct at 250
Chan et al. 2024	MLE-bench	Model, scaffold	Attempts, hours	o1-preview 16.9 to 34.1 pct at pass@8
Wijk et al. 2024	RE-Bench, agents vs experts	Environment	Time budget	Agents 4x at 2 h; experts 2x at 32 h
Lindenbauer et al. 2025	SWE-agent, 5 model configs	Harness except context	Context policy	Masking halves cost vs raw agent; solve rate matches LLM summarisation
Wiedmann et al. 2026	4 scientific tasks, 18k runs	Configuration	Run	54 pct of variance is run-to-run
Bhat et al. 2026	LiveMCPBench, 23 reruns	Everything	Run	Scores 57.9-76.8 pct

Algorithm 1. Licensing a harness claim $\delta(h_1, h_2 \mid m, T, C, R)$.

Input: claim Q , systems (m, h_1) , (m, h_2) , tasks T , runs per cell k . 1 CLASSIFY Q as DEPLOY, ATTRIBUTE or ELICIT 2 DECLARE C before running: model settings; task slice and exclusion rule; budget unit; cap or consumption; attempts and aggregation; design budget; k 3 if $Q = \text{DEPLOY}$: run each harness under its native controls report pass rate, pass^k , spend per solved task license: h_2 beats h_1 for m as shipped, at the stated prices, and for m only 4 if $Q = \text{ATTRIBUTE}$: choose spend interval $[a, b]$ in consumed units run each harness at ≥ 3 spend levels in $[a, b]$ include ≥ 2 model families m_1, m_2 report the frontier with intervals over k runs if h_2 dominates h_1 at every level for a family: license dominance on $[a, b]$ for that family elif the frontiers cross: report the crossing else: no claim 5 if $Q = \text{ELICIT}$: raise the budget until the curve flattens or funds end; report the curve and its best point 6 in every case: if the gap is below twice the run-to-run plus task-sampling standard error, report it as unresolved, whatever its sign 7 REPORT the design budget: configurations tried and the data used to select among them

scaffolds depends on model family. Kim et al. (2025) license a task-structure claim about which architectures help where, under a matched iteration cap; an

attribution claim about coordination as such would need the paper to settle whether its overhead is in messages or in tokens, and, if in tokens, to bring that overhead inside

the comparison.

The procedure applied to the older harness literature gives the same pattern. Kapoor et al.'s re-evaluation is a frontier comparison under the attribution reading and found the simple retry baseline on the frontier. Lindenbauer et al. and Moghe and Chin report accuracy and cost together for each design, and their claims survive the procedure as stated. The many harness papers that report accuracy alone on a public leaderboard license a deployment claim for one model at undisclosed prices, which is less than they usually say.

12. Relation to prior work

Five lines of work anticipate parts of this argument, and the paper's contribution is the joint reading.

Kapoor et al. (2024) argued that agent evaluations must be cost-controlled, introduced the accuracy-cost Pareto view for agents, and separated the benchmarking needs of model developers from those of downstream developers. HAL then built standardised infrastructure for evaluating across models, scaffolds and benchmarks and documented, among other things, that reasoning-effort settings do not mean the same thing across providers (Kapoor et al., 2025). This paper extends that line in two places: it argues that a matched ceiling is not cost control, using the 2026 harness studies as evidence, and it adds the attribution question as distinct from both the model-developer and downstream-developer questions.

Zhang et al. (2026a) argue that the harness is the binding constraint in a regime of comparable frontier models, propose a disclosure card covering execution, tools, context, scheduling, observability, verification and governance, and propose a factorial variance-decomposition protocol. The card's execution layer already asks for step and task budgets to be disclosed. This paper agrees with the disclosure and factorial proposals and argues that they are not sufficient for attribution, because disclosing a cap is not the same as matching consumption: a factorial grid under a shared cap reports harness variance that mixes design and spend, and a variance ratio computed over models chosen for comparability will be large partly by construction. Their own appendix supplies the cost data that makes the point.

Gorinova et al. (2026) argue that coding benchmarks conflate model, harness and environment and lack component-level signal. Zhang et al. (2026b) formalise a double confound of scaffold ownership and scorer validity. Wiedmann et al. (2026) argue that agents should be evaluated as configurable systems and measure how much of their outcome variance is noise. This paper's seven-part control set is a superset of the factors those papers name, organised by where two studies can differ rather than by where a single study can go wrong.

The machine-learning methodology literature of the late 2010s - Henderson et al. (2017), Lucic et al. (2017),

Melis et al. (2017), Dodge et al. (2019), Musgrave et al. (2020), Bouthillier et al. (2021), Agarwal et al. (2021) - established that comparisons must be made at matched tuning budget, across seeds, with intervals. This paper's claim is that the agent harness literature is the same situation with a new kind of budget: run-time spend, which a harness controls, sits alongside design-time spend, which a tuning procedure controls, and both have to be declared.

Finally, the author's earlier analysis of budget awareness in agents argued that published allocation gains are keyed to signals measured after the fact and that no forecast of an agent's own spend is scored against its run-to-run variance (Mahendrakar, 2026). That paper concerned an agent's knowledge of its own consumption; this one concerns an evaluator's control of it. The two meet in the observation that the quantity a harness is best placed to change is the one evaluations least often hold fixed.

To the best of the author's knowledge, no published work states the cap-versus-consumption distinction for harness comparisons, reads the 2026 controlled harness studies as estimating different contrasts, or maps the deployment, attribution and elicitation questions onto distinct control sets. Each component is anticipated somewhere above; the claim of novelty is for their combination applied to harness evaluation.

13. What would settle the open part

Four studies would settle most of what this paper leaves open, and none requires a new benchmark.

First, a matched-consumption re-run of an existing factorial. Take the grid of Zhang et al. (2026a) or Starace (2026), and for each harness run at three or more levels of consumed tokens, enforced by truncation or by a dollar cap the harness cannot exceed, rather than by a step cap. If the full harness's advantage for GLM-5.1 survives at the minimal harness's spend, the gain is a design gain; if it vanishes, it was a purchase. Either answer is informative, and the existing grid shows that both directions occur.

Second, a control-swap study on one benchmark: the same models and harnesses run once under native controls, once under a shared step cap, once under matched consumption, and once with spend unbounded. This would measure directly how much the control set moves the harness effect, which Figure 1 can only suggest because its studies differ in everything at once.

Third, a cross-family replication of a harness ranking with enough runs to estimate the interaction term with an interval. Starace's family-conditional result is the strongest interaction finding available and rests on one model per non-Anthropoc family. Two or more models per family, with the interaction estimated rather than inferred from reversals, would settle whether the family dependence is general.

Fourth, a design-budget curve for harness optimisers: performance of the best harness found as a function of

the number of configurations evaluated and the size of the selection set, in the form Dodge et al. (2019) proposed for hyperparameter search. Gideoni et al. (2026) and Wang et al. (2026) suggest that selection noise and overfitting to the selection set are large; a curve would say how large and at what budget they stop dominating.

14. What is not known

The size of the harness effect at matched consumption is not known for any model on any public benchmark. No located study reports harness comparisons at matched realised spend across a spend interval.

How much of the reported spread in harness effects is due to the control set, as opposed to benchmark, model and harness choice, is not known. The three controlled studies and one observational audit differ in all of these at once.

Whether the interaction between model family and harness is a stable feature or a property of the specific models and scaffolds tested is not known. The strongest evidence rests on one or two models per family.

The run-to-run share of variance for agents on the large public coding and terminal benchmarks is not known with an interval. The 54 percent figure comes from four scientific tasks under one agent, and the bimodality result from one computer-use policy.

How to convert between budget units - steps, turns, tokens, dollars, wall-clock time - so that a cap in one can be compared with a cap in another is not known in general. Prices change, providers account for cached and reasoning tokens differently, and the API-reported counts are themselves unreliable for at least one provider (Tran and Kiela, 2026).

What the design budget of any widely used manually built harness was is not known. It is not reported.

15. The case against this analysis

A sceptic has four good objections.

The first is that the deployment question is the only one that matters, and Vats and Golev answered it correctly. Users run harnesses as shipped; nobody runs a harness at matched consumption. On this view the attribution question is an academic indulgence. The reply is that method papers routinely make attribution claims - that a verifier, a planning step or a context policy is responsible for a gain - and that those claims guide what the next harness includes. If the gain was bought with spend, the next harness will include a component whose value was the extra calls, and a cheaper retry loop would have done as well, as Kapoor et al. (2024) found. Still, the objection is right that for a practitioner choosing between two products, native controls with spend reported are sufficient, and this paper does not claim otherwise.

The second is that matched consumption is unattainable. Providers do not expose reliable token counts for

reasoning, caps are advisory, and enforcing a dollar budget by truncation changes the harness's behaviour in ways that are themselves a harness effect. Tran and Kiela (2026) matched the requested budget because nothing else could be set. This objection is partly correct. The reply is that dollars charged are observable after the fact even when tokens are not, that truncation at a hard dollar cap is a well-defined condition even if it is a harsh one, and that a frontier plotted against realised spend does not require matching in advance: it requires running each harness at several budget levels and reporting where each lands. Starace (2026) and Moghe and Chin (2026) already report realised cost per correct answer for every cell; the step from there to a frontier is more cells, not a new instrument.

The third is that the paper's evidence base is thin where it matters most. The three controlled harness studies are recent, small - fifty, one hundred, and 139 tasks - and in several cases unreviewed preprints. The interaction and noise-floor evidence comes from adjacent settings rather than from the harness studies themselves. This objection is correct, and it applies to the harness literature generally, not only to this reading of it. It argues for weighting the structural points - a cap is not a spend; a claim with no declared budget excludes nothing - above the empirical ones, and those structural points do not depend on any one study's numbers.

The fourth is that the paper's prescriptions are what careful experimenters already do, and that Zhang et al. (2026a), Starace (2026) and Wiedmann et al. (2026) are already doing most of them. That is largely true, and it is the strongest reason to think the problem is tractable. The contribution claimed here is narrower than a new method: it is the observation that the careful studies disagree because they controlled different things, and that a reader who pools them, or who cites one as the size of the harness effect, is making an error the studies themselves do not make.

The reader should judge this paper on Sections 3 and 4, which rest on the published designs and resource tables of the cited studies and can be checked against them directly. The thresholds in Algorithm 1 and the three-way mapping in Table 2 are proposals whose value depends on whether they make claims easier to check, which only use can show.

16. Conclusion

A harness effect is a contrast, and a contrast is defined by what it holds fixed. The 2026 controlled studies of agent harnesses held different things fixed - native budget controls, a shared step cap, tasks and attempts with spend free - and reported within-model gaps from 2 to 28 points. Those are not draws from one distribution. Because agent accuracy can be bought with attempts, time and retries, and because the harness is the component that decides how much of a budget is spent, a harness claim that does not declare its budget excludes no observation, and a claim made under a matched ceiling cannot separate design from spend. Because

model and harness interact, a harness effect estimated over one grid of models does not carry to another. Because run-to-run variance is of the same order as many reported gaps, a single run per cell resolves less than its point estimate suggests. The remedy is not a single correct control. It is to say which question a comparison answers - which system to deploy, what a design change is responsible for, or how much a model can do - and to hold fixed what that question requires.

References

- Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A., & Bellemare, M. G. (2021). Deep Reinforcement Learning at the Edge of the Statistical Precipice. *arXiv:2108.13264*. doi:10.48550/arXiv.2108.13264. [PREPRINT]
- Bean, A. M., Kearns, R. O., Romanou, A., Hafner, F. S., Mayne, H., Batzner, J., et al. (2025). Measuring what Matters: Construct Validity in Large Language Model Benchmarks. *arXiv:2511.04703*. doi:10.48550/arXiv.2511.04703. [PREPRINT]
- Bhat, V., Vaghasiya, J., Mohsin, M. A., & Aali, A. (2026). Benchmarking the Benchmarks: A Validity Audit of Tool-Calling Evaluation. *arXiv:2607.02577*. doi:10.48550/arXiv.2607.02577. [PREPRINT]
- Bouthillier, X., Delaunay, P., Bronzi, M., Trofimov, A., Nichyporuk, B., Szeto, J., et al. (2021). Accounting for Variance in Machine Learning Benchmarks. *arXiv:2103.03098*. doi:10.48550/arXiv.2103.03098. [PREPRINT]
- Brown, B., Juravsky, J., Ehrlich, R., Clark, R., Le, Q. V., Re, C., & Mirhoseini, A. (2024). Large Language Monkeys: Scaling Inference Compute with Repeated Sampling. *arXiv:2407.21787*. doi:10.48550/arXiv.2407.21787. [PREPRINT]
- Chan, J. S., Chowdhury, N., Jaffe, O., Aung, J., Sherburn, D., Mays, E., et al. (2024). MLE-bench: Evaluating Machine Learning Agents on Machine Learning Engineering. *arXiv:2410.07095*. doi:10.48550/arXiv.2410.07095. [PREPRINT]
- Chen, L., Davis, J. Q., Hanin, B., Bailis, P., Stoica, I., Zaharia, M., & Zou, J. (2024). Are More LLM Calls All You Need? Towards Scaling Laws of Compound Inference Systems. *arXiv:2403.02419*. doi:10.48550/arXiv.2403.02419. [PREPRINT]
- Dastidar, A. (2026). Where Does Agent Reliability Come From? A Cross-Benchmark Decomposition of Verification Loops, Specialist Models, and Scaffolding in a Production Enterprise Agent. *arXiv:2607.17044*. doi:10.48550/arXiv.2607.17044. [PREPRINT]
- Davidson, T., Denain, J.-S., Villalobos, P., & Bas, G. (2023). AI capabilities can be significantly improved without expensive retraining. *arXiv:2312.07413*. doi:10.48550/arXiv.2312.07413. [PREPRINT]
- Dodge, J., Gururangan, S., Card, D., Schwartz, R., & Smith, N. A. (2019). Show Your Work: Improved Reporting of Experimental Results. *arXiv:1909.03004*. doi:10.48550/arXiv.1909.03004. [PREPRINT]
- Erol, M. H., El, B., Suzgun, M., Yuksekgonul, M., & Zou, J. (2025). Cost-of-Pass: An Economic Framework for Evaluating Language Models. *arXiv:2504.13359*. doi:10.48550/arXiv.2504.13359. [PREPRINT]
- Fan, Z., Vasilevski, K., Lin, D., Chen, B., Chen, Y., Zhong, Z., et al. (2025). SWE-Effi: Re-Evaluating Software AI Agent System Effectiveness Under Resource Constraints. *arXiv:2509.09853*. doi:10.48550/arXiv.2509.09853. [PREPRINT]
- Gideoni, Y., Risi, S., & Gal, Y. (2026). Simple Baselines are Competitive with Code Evolution. *arXiv:2602.16805*. doi:10.48550/arXiv.2602.16805. [PREPRINT]
- Gorinova, M. I., Baker, M., Heineike, A., Shaposhnikov, M., Willoughby, R., & Knox, D. (2026). Position: Coding Benchmarks Are Misaligned with Agentic Software Engineering. *arXiv:2606.17799*. doi:10.48550/arXiv.2606.17799. [PREPRINT]
- Guo, J., Hao, Z., Wang, C., Fan, C., Luo, T., Li, H., et al. (2026). From Question Answering to Task Completion: A Survey on Agent System and Harness Design. *arXiv:2606.20683*. doi:10.48550/arXiv.2606.20683. [PREPRINT]
- Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., & Meger, D. (2017). Deep Reinforcement Learning that Matters. *arXiv:1709.06560*. doi:10.48550/arXiv.1709.06560. [PREPRINT]
- Hu, S., Lu, C., & Clune, J. (2024). Automated Design of Agentic Systems. *arXiv:2408.08435*. doi:10.48550/arXiv.2408.08435. [PREPRINT]
- Jiang, M., Ruan, Y., Lastras, L., Kapanipathi, P., & Hashimoto, T. (2025). Putting It All into Context: Simplifying Agents with LCLMs. *arXiv:2505.08120*. doi:10.48550/arXiv.2505.08120. [PREPRINT]
- Kapoor, S., Stroebl, B., Siegel, Z. S., Nadgir, N., & Narayanan, A. (2024). AI Agents That Matter. *arXiv:2407.01502*. doi:10.48550/arXiv.2407.01502. [PREPRINT]
- Kapoor, S., Stroebl, B., Kirgis, P., Nadgir, N., Siegel, Z. S., Wei, B., et al. (2025). Holistic Agent Leaderboard: The Missing Infrastructure for AI Agent Evaluation. *arXiv:2510.11977*. doi:10.48550/arXiv.2510.11977. [PREPRINT]
- Kim, Y., Gu, K., Park, C., Park, C., Schmidgall, S., Heydari, A. A., et al. (2025). Towards a Science of Scaling Agent Systems. *arXiv:2512.08296*. doi:10.48550/arXiv.2512.08296. [PREPRINT]
- Kozyrev, M., Kozyrev, A., & Podkopaev, A. (2026). Skill Issue: Lessons from Optimizing Repository SKILLS for Coding Agents. *arXiv:2609.12742*. doi:10.48550/arXiv.2609.12742. [PREPRINT]
- Kwa, T., West, B., Becker, J., Deng, A., Garcia, K., Hasin, M., et al. (2025). Measuring AI Ability to Complete Long Software Tasks. *arXiv:2503.14499*. doi:10.48550/arXiv.2503.14499. [PREPRINT]
- Lindenbauer, T., Slinko, I., Felder, L., Bogomolov, E., & Zharov, Y. (2025). The Complexity Trap: Simple Observation Masking Is as Efficient as LLM Summarization for Agent Context Management. *arXiv:2508.21433*. doi:10.48550/arXiv.2508.21433. [PREPRINT]
- Lip, E. L. C., Moraski, B., Knappe, T., Xiong, L., Gharat, S., Mari, A., & Bercovich, I. (2026). What Makes a Terminal-Bench Task Hard? Separating Genuine Hardness from Fake-Hardness on an Adjudicated Agentic Corpus. *arXiv:2609.26826*. doi:10.48550/arXiv.2609.26826. [PREPRINT]
- Liu, F., Liu, Y., Sun, R., Luo, L., & Guo, S. (2026). Coding Agents Have Converged: Why the SWE-bench Leaderboard Can No Longer Order Its Top Entries, and What to Measure Instead. *International Conference on Advanced Data Mining and Applications (ADMA) 2026*. *arXiv:2609.17394*. doi:10.48550/arXiv.2609.17394.
- Lucic, M., Kurach, K., Michalski, M., Gelly, S., & Bousquet, O. (2017). Are GANs Created Equal? A Large-Scale Study. *arXiv:1711.10337*. doi:10.48550/arXiv.1711.10337. [PREPRINT]
- Mahendrakar, P. (2026). Three Things Called Budget Awareness: Observability, Forecasting and Allocation in LLM Agents, Why Every Published Allocation Gain Is Keyed to a Signal Measured After the Fact, and the Run-to-Run Variance No Forecast Is Scored Against. *Life of Research*. doi:10.5281/zenodo.22884949.
- Melis, G., Dyer, C., & Blunsom, P. (2017). On the State of the Art of Evaluation in Neural Language Models. *arXiv:1707.05589*. doi:10.48550/arXiv.1707.05589. [PREPRINT]
- Mialon, G., Fourier, C., Swift, C., Wolf, T., LeCun, Y., & Scialom, T. (2023). GAIA: a benchmark for General AI Assistants. *arXiv:2311.12983*. doi:10.48550/arXiv.2311.12983. [PREPRINT]
- Miller, E. (2024). Adding Error Bars to Evals: A Statistical Approach to Language Model Evaluations. *arXiv:2411.00640*. doi:10.48550/arXiv.2411.00640. [PREPRINT]
- Moghe, K., & Chin, P. (2026). Cost-Effective Agent Harnesses for Abstract Reasoning and Generalization on ARC-AGI-1. *arXiv:2607.06764*. doi:10.48550/arXiv.2607.06764. [PREPRINT]
- Musgrave, K., Belongie, S., & Lim, S.-N. (2020). A Metric Learning Reality Check. *arXiv:2003.08505*. doi:10.48550/arXiv.2003.08505. [PREPRINT]

- Qin, Z., Lu, Y., Wang, Z. A., & Wang, K. (2026). StateM: Reaching 95.3% Raw Accuracy, or a \$15 Frontier Run, on Terminal-Bench 2.1 via Harness Scaling. arXiv:2608.15089. doi:10.48550/arXiv.2608.15089. [PREPRINT]
- Sahu, B., & Pandey, S. (2026). Teach it to stop, not just to click. arXiv:2607.17136. doi:10.48550/arXiv.2607.17136. [PREPRINT]
- Shen, J., Bai, H., Zhang, L., Zhou, Y., Setlur, A., Tong, S., et al. (2025). Thinking vs. Doing: Agents that Reason by Scaling Test-Time Interaction. arXiv:2506.07976. doi:10.48550/arXiv.2506.07976. [PREPRINT]
- Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language Agents with Verbal Reinforcement Learning. arXiv:2303.11366. doi:10.48550/arXiv.2303.11366. [PREPRINT]
- Singh, S., Nan, Y., Wang, A., D'Souza, D., Kapoor, S., Ustun, A., et al. (2025). The Leaderboard Illusion. arXiv:2504.20879. doi:10.48550/arXiv.2504.20879. [PREPRINT]
- Snell, C., Lee, J., Xu, K., & Kumar, A. (2024). Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters. arXiv:2408.03314. doi:10.48550/arXiv.2408.03314. [PREPRINT]
- Starace, J. (2026). Scaffold Effects on GAIA: A Controlled Comparison. arXiv:2606.08529. doi:10.48550/arXiv.2606.08529. [PREPRINT]
- Thangarajah, K., Chen, B., & Hassan, A. E. (2026). DCAS: Decoupling CLI Agent Scaffolding to Internalize Planning across Scaffolds. arXiv:2608.06113. doi:10.48550/arXiv.2608.06113. [PREPRINT]
- Tran, D., & Kiela, D. (2026). Single-Agent LLMs Outperform Multi-Agent Systems on Multi-Hop Reasoning Under Equal Thinking Token Budgets. arXiv:2604.02460. doi:10.48550/arXiv.2604.02460. [PREPRINT]
- Vats, N., & Golev, O. (2026). The Scaffold Effect in Coding Agents: Harness Choice as a Hidden Variable in Coding-Agent Evaluation. arXiv:2607.22585. doi:10.48550/arXiv.2607.22585. [PREPRINT]
- Wang, N., Hu, X., Liu, P., Zhu, H., Hou, Y., Huang, H., et al. (2025). Efficient Agents: Building Effective Agents While Reducing Cost. arXiv:2508.02694. doi:10.48550/arXiv.2508.02694. [PREPRINT]
- Wang, W., Kattakinda, P., & Feizi, S. (2026). Do Agent Optimizers Compound? A Continual-Learning Evaluation on Terminal-Bench 2.0. arXiv:2607.14004. doi:10.48550/arXiv.2607.14004. [PREPRINT]
- Weinberger, S., & Hozes, A. (2026). Prompt-Induced Waste in Coding Agents: Reasoning, Effort, Harness Design, and End-to-End Cost. arXiv:2608.01347. doi:10.48550/arXiv.2608.01347. [PREPRINT]
- Wiedmann, L., Girrbach, L., Schmid, C., & Akata, Z. (2026). Agents Are Systems, Not Models: Rethinking Agentic Evaluation. arXiv:2610.01618. doi:10.48550/arXiv.2610.01618. [PREPRINT]
- Wijk, H., Lin, T., Becker, J., Jawhar, S., Parikh, N., Broadley, T., et al. (2024). RE-Bench: Evaluating frontier AI R&D capabilities of language model agents against human experts. arXiv:2411.15114. doi:10.48550/arXiv.2411.15114. [PREPRINT]
- Wu, Y., Zhang, J., Shi, J., Lei, X., Gu, Q., Zhang, Y., et al. (2026). HarnessDev: Can LLMs Create and Evolve Their Own Agent Harness? arXiv:2609.01437. doi:10.48550/arXiv.2609.01437. [PREPRINT]
- Xia, C. S., Deng, Y., Dunn, S., & Zhang, L. (2024). Agentless: Demystifying LLM-based Software Engineering Agents. arXiv:2407.01489. doi:10.48550/arXiv.2407.01489. [PREPRINT]
- Xu, J., Koesdwiady, A., Bei, S., Han, Y., Huang, B., Wang, D., et al. (2026). Rethinking the Value of Multi-Agent Workflow: A Strong Single Agent Baseline. arXiv:2601.12307. doi:10.48550/arXiv.2601.12307. [PREPRINT]
- Yang, J., Jimenez, C. E., Wettig, A., Lieret, K., Yao, S., Narasimhan, K., & Press, O. (2024). SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. arXiv:2405.15793. doi:10.48550/arXiv.2405.15793. [PREPRINT]
- Yang, Z., Wang, S., Fu, K., He, W., Xiong, W., Liu, Y., et al. (2025). Kimi-Dev: Agentless Training as Skill Prior for SWE-Agents. arXiv:2509.23045. doi:10.48550/arXiv.2509.23045. [PREPRINT]
- Yao, S., Shinn, N., Razavi, P., & Narasimhan, K. (2024). tau-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. arXiv:2406.12045. doi:10.48550/arXiv.2406.12045. [PREPRINT]
- Zhang, H., Cui, Z., Chen, J., Wang, X., Zhang, Q., Wang, Z., et al. (2025). Stop Overvaluing Multi-Agent Debate – We Must Rethink Evaluation and Embrace Model Heterogeneity. arXiv:2502.08788. doi:10.48550/arXiv.2502.08788. [PREPRINT]
- Zhang, Y., Wang, J., Ge, Y., Xu, W., Hamm, J., & Reddy, C. K. (2026a). Stop Comparing LLM Agents Without Disclosing the Harness. arXiv:2605.23950. doi:10.48550/arXiv.2605.23950. [PREPRINT]
- Zhang, Y., Motaali, S., Dinh, V. P., Piroutiniya, A., Lopez de Vergara, J. E., de Pedro, L., et al. (2026b). The Double Measurement Confound in Agent Benchmarks: De-Scaffolding, Ground-Truth Scoring, and Reliability Beyond the Mean. arXiv:2609.09218. doi:10.48550/arXiv.2609.09218. [PREPRINT]
- Zhang, H., Xu, Q., Li, Z., Zhang, L., Jiang, P., Zhang, Y., & McAuley, J. (2026c). Masking Stale Observations Helps Search Agents – Until It Doesn't: A Regime Map and Its Mechanism. arXiv:2606.00408. doi:10.48550/arXiv.2606.00408. [PREPRINT]
- Zhou, A., Yan, K., Shlapentokh-Rothman, M., Wang, H., & Wang, Y.-X. (2024). Language Agent Tree Search Unifies Reasoning Acting and Planning in Language Models. arXiv:2310.04406. doi:10.48550/arXiv.2310.04406. [PREPRINT]
- Zhu, Q., Chen, T., Lu, S., Ji, L., Zhao, R., Ma, M., et al. (2026). Pull Requests as a Training Signal for Repo-Level Code Editing. International Conference on Machine Learning (ICML) 2026. arXiv:2602.07457. doi:10.48550/arXiv.2602.07457.
- Zhu, Y., Jin, T., Pruksachatkun, Y., Zhang, A., Liu, S., Cui, S., et al. (2025). Establishing Best Practices for Building Rigorous Agentic Benchmarks. arXiv:2507.02825. doi:10.48550/arXiv.2507.02825. [PREPRINT]