

Harness vs model under matched token caps: how much of an agent coding score is the harness?

IRSI research note, 2026-10-07. Check level: CROSS-CHECKED for table 1 (scope in section 5); other tables SELF-CHECKED.

Summary

- **Setup.** The same 30 exact-scored coding tasks (Exercism, 5 languages) were run through 3 public agent harnesses (aider, mini-swe-agent, OpenHands) with 3 models (Gemini 3.8 Flash, Kimi K2.5, gpt-oss-120b). Each harness was run unmodified, with its default agent; aider’s launch flags (diff edit format, auto-test) are listed in section 4.
- **In this 3x3 grid, the harness explains more of the between-cell differences than the model at every token cap** (harness minus model share +0.22 [+0.04, +0.36] at 200k). This depends on including OpenHands: with aider and mini-swe-agent only, the model share is larger at 100k and 200k (0.34 against 0.12).
- **The budget changes the answer.** With a 20k-token cap per task, the harness explains 83% of between-cell differences and OpenHands solves 0% of tasks with every model. With 200k tokens, most cells are near the ceiling, and the largest point estimate is the model x harness interaction, at 48%.
- **The model ranking flips with the harness.** Gemini 3.8 Flash, the model Artificial Analysis rates strongest on coding (76.3, against 46.8 and 30.4; AA’s High or Reasoning settings, while the runs used each harness’s default reasoning setting), ranks first under aider (100%) and last overall at 200k (80% mean), because OpenHands solves 47% with it.
- **Robustness.** On a larger 100-task grid of the two open models, the harness share is 43% to 86% and exceeds the model share at 10k to 50k. At 100k and 200k the difference is not significant (harness minus model +0.10 [-0.15, +0.31] and +0.09 [-0.25, +0.43]).
- **Instrument.** One run per task/model/harness at a 200k stopping threshold reconstructs outcomes at lower thresholds by grading the workspace before the first call whose prior cumulative token use is already at or above the threshold. A permitted call may overshoot the threshold. Section 5 reports comparisons with real capped runs.

1. Method

- **Tasks.** 163 non-Java Exercism tasks from the Aider polyglot set had a passing reference solution and a failing stub under our grader; they were frozen as eligible before any main run. The core grid uses 30 of them (6 per language); the open-model grid uses 100.
- **Isolation.** Every run is a fresh rootless container with no network, all capabilities dropped and no credential mounts. The only outside channel is a model proxy on a unix socket. The proxy pins each run to its model, enforces the per-run token cap and logs every call.
- **Grading.** A trusted copy of the original exercise receives only the declared solution files, so test edits have no effect; unit tests run in a fresh no-network container. Pass or fail is the test exit code.
- **Token cap definition.** Tokens are cumulative per task: prompt plus completion on every call, so context re-sent on each call counts again and cached tokens count in full. This favors harnesses that make few calls (aider: median 7k to 31k tokens per task).
- **Cap sweep.** The outcome at cap B is the grade of the workspace at the first model call that would start at or above B tokens. One 200k run per cell gives outcomes at 10k, 20k, 50k, 100k and 200k. An entire call is allowed whenever prior consumption is below B, so its response and resulting workspace changes may occur after consumption exceeds B.
- **Analysis.** Two-way split of the cell means into model, harness and interaction shares; language-stratified paired task bootstrap (2,000 resamples) for intervals; full task-level ANOVA; permutation test over harness labels (Holm across caps); pairwise contrasts; a logistic model with task fixed effects as a sensitivity check.
- **Versions.** aider-chat 0.86.2 and mini-swe-agent 2.4.6 (litellm 1.81.10); openhands-sdk 1.53.0 and openhands-tools 1.53.0 (litellm 1.104.0) in a separate environment. Aider used the diff edit format for every model.

- **Runs.** 1,144 result rows and 12,227 model calls: a 15-task pilot, the 30-task core grid, 135 real 20k-capped reruns, a 70-task open-model extension and a second seed for the open models. 4 runs were rerun under a frozen rerun-once policy for upstream errors; no cell exceeded 5 runs with upstream errors.

2. Results

Table 1. Core 3x3 grid, n = 30 tasks, shares of between-cell variance

Point estimates are bias-corrected; brackets are 95% bootstrap CIs (percentile intervals of the raw shares). The permutation p is for harness plus interaction, Holm-corrected; 0.0025 is its floor with 2,000 permutations. The last column is harness share minus model share.

token cap per task	model	harness	interaction	perm p	harness range, Gemini 3.8 Flash	range, Kimi K2.5	range, gpt-oss-120b	harness minus model
10k	0.18 [0.12, 0.24]	0.64 [0.52, 0.73]	0.19 [0.14, 0.26]	0.0025	93 pts [83, 100]	80 [67, 93]	40 [23, 57]	+0.45 [+0.29, +0.61]
20k	0.07 [0.03, 0.11]	0.83 [0.74, 0.88]	0.11 [0.07, 0.18]	0.0025	97 [90, 100]	87 [77, 97]	60 [47, 77]	+0.75 [+0.64, +0.84]
50k	0.23 [0.17, 0.29]	0.53 [0.45, 0.59]	0.24 [0.18, 0.32]	0.0025	93 [83, 100]	7 [3, 17]	70 [53, 87]	+0.29 [+0.18, +0.40]
100k	0.15 [0.09, 0.23]	0.52 [0.36, 0.63]	0.33 [0.23, 0.50]	0.0025	77 [63, 90]	3 [0, 17]	33 [17, 53]	+0.36 [+0.18, +0.50]
200k	0.14 [0.07, 0.26]	0.38 [0.22, 0.50]	0.48 [0.33, 0.66]	0.0025	53 [37, 70]	10 [3, 20]	10 [3, 23]	+0.22 [+0.04, +0.36]

Table 2. Core cell pass rates (%), n = 30 per cell, Wilson 95% intervals

model	aider @20k	mini @20k	OpenHands @20k	aider @200k	mini @200k	OpenHands @200k	median tokens used (aider / mini / OpenHands)
Gemini 3.8 Flash	96.7 [83, 99]	30.0 [17, 48]	0.0 [0, 11]	100.0 [89, 100]	93.3 [79, 98]	46.7 [30, 64]	11k / 162k / 207k
Kimi K2.5	86.7 [70, 95]	80.0 [63, 90]	0.0 [0, 11]	90.0 [74, 97]	100.0 [89, 100]	93.3 [79, 98]	9k / 39k / 89k
gpt-oss-120b	60.0 [42, 75]	36.7 [22, 54]	0.0 [0, 11]	86.7 [70, 95]	93.3 [79, 98]	83.3 [66, 93]	31k / 74k / 177k

At n = 30 a cell's 95% interval is about ± 15 to 18 points mid-range. At 200k: model means Gemini 80.0, Kimi 94.4, gpt-oss 87.8; harness means aider 92.2, mini 95.6, OpenHands 74.4. For Gemini at 200k, aider minus OpenHands is +53 points [37, 70] and mini minus OpenHands +47 [30, 63].

Table 3. Open-model 2x3 grid (Kimi K2.5, gpt-oss-120b), n = 100 tasks

Model, harness and interaction point estimates are bias-corrected; their brackets are raw-share bootstrap CIs, as in table 1.

cap	model	harness	interaction	Kimi harness range	gpt-oss harness range	harness minus model
10k	0.24 [0.17, 0.30]	0.61 [0.53, 0.69]	0.15 [0.12, 0.18]	79 [71, 87]	45 [35, 55]	+0.37 [+0.23, +0.52]
20k	0.09 [0.05, 0.14]	0.86 [0.78, 0.92]	0.05 [0.03, 0.09]	81 [74, 89]	60 [50, 69]	+0.77 [+0.64, +0.86]
50k	0.29 [0.20, 0.40]	0.53 [0.41, 0.62]	0.18 [0.10, 0.28]	20 [12, 29]	65 [56, 74]	+0.23 [+0.03, +0.42]
100k	0.33 [0.19, 0.47]	0.43 [0.29, 0.55]	0.25 [0.14, 0.38]	13 [8, 21]	42 [33, 54]	+0.10 [-0.15, +0.31]
200k	0.38 [0.15, 0.58]	0.47 [0.28, 0.65]	0.15 [0.03, 0.41]	15 [9, 22]	20 [11, 29]	+0.09 [-0.25, +0.43]

Pass rates at 200k: Kimi aider 84%, mini 99%, OpenHands 89%; gpt-oss aider 82%, mini 88%, OpenHands 68%.

Table 4. Leave OpenHands out (aider and mini-swe-agent only): raw shares model / harness / interaction [95% CI]

cap	core 3x2, n = 30			open 2x2, n = 100		
10k	0.47	[0.31, 0.60]	/ 0.36 [0.22, 0.52] / 0.17	0.66	[0.50, 0.78]	/ 0.29 [0.17, 0.44] / 0.05
20k	0.33	[0.18, 0.50]	/ 0.42 [0.19, 0.59] / 0.26	0.75	[0.53, 0.91]	/ 0.18 [0.05, 0.37] / 0.06
50k	0.19	[0.06, 0.47]	/ 0.41 [0.20, 0.55] / 0.40	0.59	[0.18, 0.90]	/ 0.06 [0.00, 0.21] / 0.35
100k	0.34	[0.02, 0.86]	/ 0.12 [0.00, 0.42] / 0.55	0.43	[0.03, 0.82]	/ 0.33 [0.05, 0.68] / 0.24
200k	0.34	[0.04, 0.65]	/ 0.12 [0.00, 0.31] / 0.55	0.24	[0.01, 0.61]	/ 0.64 [0.28, 0.86] / 0.12

These are raw shares, so they are only roughly comparable with table 1. The intervals are wide; without OpenHands, harness does not beat model at most caps.

Table 5. Share of all task-level variance (full ANOVA), core grid

MedicalHarness (2610.05778) decomposes per-seed task-averaged outcomes and averages the resulting shares across four environments: model 57.5%, harness 12.4%, interaction 11.9% and run-to-run variation 18.2%. Its unit of analysis differs from our task-level ANOVA, so these percentages are not directly comparable.

cap	task	model	harness	model x harness	task x model	task x harness	residual
20k	0.09	0.04	0.45	0.06	0.07	0.12	0.17
200k	0.19	0.03	0.08	0.10	0.13	0.12	0.35

At 200k, the task main effect accounts for 19% of total variance and the residual for 35%. With one primary observation per task/model/harness combination, the residual combines task-by-model-by-harness interaction and run noise; this ANOVA does not separate them. The between-cell sum of squares is 0.21 at 200k, against 1.22 at 20k (squared pass-rate units), so the table 1 shares at 200k describe smaller differences.

3. Observed vs inferred

Observed - OpenHands’ first call uses 6.9k to 10.0k input tokens (median 7.3k, n = 381 runs). In the core grid, none of its graded workspaces passed at the 10k or 20k checkpoints. - Gemini 3.8 Flash with OpenHands made many short calls (median about 48 output tokens per call) and hit its token cap in 57 of 60 runs (28 of 30 in the core grid at 200k). - The OpenAI-compatible route used for Gemini returns its encrypted reasoning (`reasoning_details`). The OpenHands event log never stores it; the mini-swe-agent trajectory does.

Inferred - The Gemini x OpenHands deficit may be partly a route artifact: OpenHands drops Gemini’s reasoning state between turns on this route, so Gemini may re-reason every turn. This is untested; no run used Gemini’s native API. All three harnesses saw the model under a generic OpenAI-compatible name (section 4), which is a plausible reason the reasoning state was not carried on this route. - If so, part of the core interaction share is “harness x provider route”, not harness x model. The open-model grid (no Gemini) shows a higher point estimate for harness share at every cap, significant at 10k to 50k only, so the 100k and 200k headline partly rests on the Gemini x OpenHands cell.

Unknown - Uncapped behavior: the 200k cap still binds for OpenHands and mini-swe-agent with Gemini (median 207k and 162k tokens). - Transfer to repository-level tasks (SWE-bench, Terminal-Bench). - How much memorization of the public Exercism tasks lifts levels.

4. Limitations

- **Harness packages as configured.** Aider used the diff edit format for every model (its own leaderboard picks a format per model). Sampling and reasoning settings are each harness’s defaults. The estimand is “harness package as configured”, not harness architecture alone. All three harnesses were given the model under a generic OpenAI-compatible name, so model-specific settings built into them (aider’s per-model edit

formats and prompts, any provider-specific handling of reasoning state in OpenHands or litellm) were not applied. Aider was launched with `-auto-test` (it reruns the tests and retries up to 3 times) and with linting off.

- **Home benchmark.** The tasks come from Aider’s own polyglot benchmark, which aider’s prompts and edit formats were developed against, while OpenHands is built mainly for repository-level tasks. This may favor aider and count against OpenHands, the two harnesses that drive the harness share.
- **Dependence on OpenHands.** “Harness explains more than model” is a property of this 3x3 grid that holds only with OpenHands included (table 4), not a general law.
- **Low caps.** All 90 core OpenHands trajectories made two successful model calls before the 10k checkpoint and three before the 20k checkpoint. None produced a passing graded workspace at those checkpoints. These results describe the configured workflows under our stopping policy; they do not establish that the first prompt prevents a second call.
- **Token accounting.** The cap counts re-sent context and ignores caching, which favors few-call harnesses.
- **Small closed-model arm.** $n = 30$ for the 3x3 grid. Run noise was measured on the two open models only; Gemini has one seed. With one closed model, the grid cannot separate an “openness” effect from that model’s other traits.
- **Harness range.** The harness range (best minus worst harness) is biased upward by noise; the pairwise contrasts are the cleaner comparison.
- **Ceiling at 200k.** Most cells are at or above 83% at 200k, which compresses the shares.
- **Versions.** The image was built from unpinned package names; the installed versions above were recorded from the build logs after the fact.

5. Checks

check	result	bar
Independent recompute: an analysis script written by GPT-6.1 Sol (OpenAI) from the spec alone, run in a no-network sandbox	125/125 quantities agree on the final core data	all agree
Cap instrument: grade after a refused call equals the snapshot at that call	OpenHands 45/45; aider 11/11; mini 43/44	95% per harness
Real 20k-capped runs vs reconstructed 20k outcomes (different samples, $n = 15$ each)	all 9 cells within 13.3 points (2 of 15 tasks); OpenHands 0% vs 0% in all three	distributional agreement
Final grade vs cap-sweep grade at the run’s own cap	0 mismatches over the 1,005 runs capped at 200k; 1 over the 135 real 20k-capped runs (a mini-swe-agent run refused at the cap, the 43/44 case above)	0 at 200k
Grader self-test (reference passes, stub fails)	163/178 tasks	at least 90%
Run noise (seed 0 vs seed 2, open models, 180 pairs)	runs disagree 9% (200k) to 17% (100k)	expected noise in the interaction SS (about 0.006 at 200k) is small against the observed 0.102
Logistic model with task fixed effects	likelihood ratio for harness > model in all 4 fits (core and open, 20k and 200k); interaction significant except open at 20k ($p = 0.10$)	sensitivity check only

The cross-check covers the raw shares, cell rates, ranges, ANOVA shares and CIs. The bias-corrected point estimates shown (which differ from the raw shares by at most 0.012) are SELF-CHECKED. The harness-minus-model intervals and table 4 come from a separate release-check recompute on the same data (SELF-CHECKED).

The final-vs-own-cap counts were recomputed from `runs/results.jsonl` (last attempt per task/model/harness/seed/cap) and `runs/capsweep.jsonl` with [check/recompute-own-cap.py](#); [check/own-cap-comparison.json](#) records the counts, mismatch and source hashes (SELF-CHECKED).

6. Related work

At least a dozen 2025 and 2026 papers cross models with agent harnesses. - **Variance splits.** Exgentic (2602.22953; five models and five architectures overall) reports an axis decomposition on its three closed models across five architectures: model 27.8% and architecture 0.5%. Its separate additive decomposition of model–architecture cell means gives model 93.9%, architecture 0.7% and interaction 5.4%. MedicalHarness (2610.05778; 5 open models x 5 harnesses, 3 seeds: model 57.5%, harness 12.4%, interaction 11.9%, noise 18.2%) reports variance splits, but without intervals on the shares and without matched token budgets. - **Matched steps.** Stop Comparing (2605.23950) runs

a 3x3 grid on 100 SWE-bench Verified tasks with 50 matched steps and finds harness variance 7.8 times model variance, with rank reversals in 6 of 9 comparisons; context and cost are not matched and no intervals are given. - **Rank reversals and scaffold effects** are documented in HAL (2510.11977), Harness-Bench (2605.27922), Scaffold Effects on GAIA (2606.08529), the Double Measurement Confound (2609.09218), Coding Agents Have Converged (2609.17394), Beyond the Model (2609.32459) and Finding the Right Fit (2610.00917). None matches per-task token budgets. - **Closest new work.** 2609.20804 varies context-window budgets (32k to 128k) for harness components inside one harness and finds that context management matters more as the budget shrinks. It does not cross public harnesses or report variance shares. 2609.11987 compares vendor and neutral harnesses on the same model under a wall-clock cap. Claw-SWE-Bench (2606.12344) compares model changes with harness changes and reports cost. It uses shared outer turn and wall-clock limits, but does not match cumulative per-task token budgets. 2607.22585 argues that a harness’s budget controls are part of the scaffold. We instead treat the per-task token budget as an external condition, and report an externally imposed 200k stopping threshold and reconstructed lower thresholds; uncapped native behavior was not measured. SWE-Effi (2509.09853) scores 5 scaffolds x 3 models on 50 SWE-bench Verified issues by the area under the resolve-rate-vs-tokens-per-issue curve up to 2M tokens, so it also reads every budget from one run per cell. It reports no variance shares, intervals or per-cap results, and it credits an issue at budget B only from the finished run; we grade the workspace snapshot at each cap and validate that against real capped runs. - **What is new here.** We found no paper that measures model, harness and interaction shares, with intervals, as a function of a matched cumulative per-task token cap. Reading every budget from one run per cell is not new (SWE-Effi does it from finished-run token totals); our addition is grading the workspace snapshot at each cap, validated against real capped runs.

7. Who this is for

- **Leaderboards that fix one scaffold or report vendor scaffolds** (HAL, swebench.com bash-only, Terminal-Bench, Epoch’s benchmarking hub). Which harness was used, and the token budget, change the score as much as or more than which model was used. At a 20k cap the harness explains 83% of between-cell differences (the model 7%), mostly because OpenHands cannot finish any task at that cap. At 200k the harness explains 38%, the interaction 48%. Model rankings can flip by harness. The concrete suggestion is to report scores at two or more stated per-task token caps.
- **METR’s time-horizon method** (fixed scaffolds, 8M and 16M token budgets): this supports budget-matched reporting, but our budgets are far smaller, so it does not test the time-horizon results directly.
- **Failure-attribution studies** (for example 2606.05238, 2607.28802): a model’s measured failure rate depends on both the harness and the budget. Kimi K2.5 under OpenHands ranges from 0% to 93% across caps.
- **Practitioners choosing a harness for a fixed budget**, the authors of 2605.23950, whose step-matched design this complements, and the SWE-Effi authors, whose budget-AUC design this complements.

8. Cost

Total model spend was about \$50 at list prices (Gemini 3.8 Flash through an OpenAI-compatible router; Kimi K2.5 and gpt-oss-120b through Amazon Bedrock).

9. Next steps

1. Rerun the Gemini x OpenHands cell through a route that preserves Gemini’s reasoning state, to separate the harness effect from the route artifact.
2. Apply the same instrument to repository-level tasks off ceiling (a SWE-bench Verified or Terminal-Bench subset).
3. Raise the cap to 500k for the two cells where 200k still binds.