



Claude Haiku 5.5 across every effort level

VulcanBench Frontier v4 | Code quality protocol v3.24 | October 2026

Abstract

Claude Haiku 5.5 ran the 23-task VulcanBench Frontier v4 suite through Claude Code 2.1.293 to 2.1.296 on a Claude Max subscription from October 8 to 11, 2026, once per task at each of five effort levels, Low to Max, 115 runs in all, one task at a time on the owner's Mac. Code quality carries 33% of the combined score and is judged for a named human reader by Muse Spark 1.3 (Meta) and Grok 4.6 (xAI), both from labs other than Anthropic, under the same rubric, controls and gates as the other Frontier v4 reports, with a ground-truth intent-recovery probe. The combined score is 84.42 at Low, 84.03 at Medium, 88.01 at High, 91.83 at Extra-high and 92.16 at Max, over 23, 22, 23, 22 and 21 judged runs. Haiku 5.5 passes 13, 13, 17, 22 and 22 of 23 tasks from Low to Max. Code quality runs from 74.83 to 81.30. Cost per task, as Claude Code reports it, runs from \$0.84 to \$1.54; the whole sweep comes to \$138.31, between a floor of \$31.10 and a ceiling of \$155.50 for Haiku's two price tiers.

Every run finished inside the flat 3-hour task bound. Four runs carry no Code quality score and all four count in tasks passed, time and cost: three that a safety classifier stopped with an empty patch, and one whose judge probe had no valid answer. Claude Code's refusal fallback was on and no run used it. The sweep paused twice between tasks, which leaves run timings intact.

Table 1. Combined score, Code quality, runtime and tasks passed by effort

Effort	Combined (33%)	SE	Combined (20%)	Code quality	Min/task	Judged	Passed
Low	84.42	1.97	86.45	74.83	22.9	23	13/23
Medium	84.03	2.50	85.92	76.07	24.0	22	13/23
High	88.01	1.42	89.69	77.60	26.8	23	17/23
Extra-high	91.83	0.39	93.18	80.13	19.0	22	22/23
Max	92.16	0.37	93.31	81.30	27.7	21	22/23

Combined (33%) = 0.50 functional + 0.085 lint and complexity + 0.085 security + 0.33 Code quality, over the judged runs. Combined (20%) applies the prior 50/15/15/20 profile to the same Code quality scores. SE is one sample standard error across the cell's judged tasks, not judge uncertainty or a significance test. Runtime is solver wall-clock per task over every run; judging is excluded. Frontier Code quality is the reviewed layer plus intent recovery and is never compared with Routine v1 Code quality.

The Code quality protocol

Two failure modes motivated the reviewed score. The lint and complexity metric rewards compression: the maintainability index carries a lines-of-code term and per-function complexity stays low when each dense line does something different, so five statements on a line with names like `x` and `k` can outscore the same logic written for a person. And a large model reading compressed code pays almost nothing to parse it; when a frontier model was calibrated as a readability judge, it rated a squashed six-line function the same as its formatted copy. Since September 7, 2026 Code quality carries 33% of the combined score and lint and complexity and security carry 8.5% each. The prior profile is reported beside the new one on every table.

Protocol v3.24 is the v3.23 protocol applied to this population. Nothing in the rubric, controls, quirk keys, gates, repeats, seed, weights or judge settings changed; both judges retook the calibration exam under v3.24 before any counted call. What changed is how the Cursor CLI that carries Grok 4.6 is pinned, described on the next page. The population froze on October 11 with 112 judgeable rows, three runs excluded and none missing.

The rubric

Every judge receives the same instructions, frozen by hash before any review. The prompt names the reader it scores for: an engineer who has never seen the code, reads it top to bottom without running it, and must make a correct change in one sitting. It tells the judge that its own ease at parsing dense code is not evidence of readability. Six dimensions are scored 0 to 4 in half steps against written anchors, each with an exact excerpt and a concrete consequence for that reader.

Sub-score	Dimension	What is assessed
Human readability	Naming	Identifiers say what things are in the task's domain
Human readability	Presentation	Statements per line, nesting, function length, how much the reader must hold at once
Human readability	Intent	Constants, thresholds and quirks explained by names or accurate comments that say why
Maintainability	Structure	Coherent responsibilities, no duplicated policy, no abstraction beyond the task's scale
Maintainability	Changeability	A named plausible change, every edit site traced, scored on locality and safety
Maintainability	Verifiability	Explicit state, failures that name what went wrong, seems to test one rule alone

The layers under the 33 points

Reviewed panel, 24 points. The six-dimension rubric above, averaged across both judges. **Intent recovery, 9 points.** Every task in this suite is a rewrite of a retired binary whose real behaviour departs from its written specification in documented ways. The judge sees only the specification and the code, never the issue text, and lists where the code departs from the specification; a separate call matches that list against the frozen answer key. The denominator is the quirks the submission actually passed tests for, so a functional failure is not punished twice; no judged Haiku 5.5 run needed the rule for a submission that passed none. **Measured maintenance, designed for 12 points, not yet built.** Until it exists the pre-registered split above is in force.

Judges and calibration

The scored panel is Muse Spark 1.3 through the Muse CLI on its Standard tier, which does not train on prompts, and Grok 4.6 through the Cursor CLI at medium effort. Both are from labs other than Anthropic, so neither grades a relative of Haiku 5.5. Every session is fresh, tools are disabled, the workspace is empty, model identity is checked per call from the CLI's own records, and solver labels are withheld.

Judge settings and versions

The judge settings are read from the original v3.3 and v3.4 protocol files, recovered from the owner's private backup, whose sha256 equal the published values (v3.3 b82da598..., v3.4 1d80e097...). Muse Spark 1.3 ran the same binary as every earlier round (sha256 4c0f960028b6..., a match). Grok 4.6 ran through Cursor CLI 2026.10.01-e373342, pinned by version for the whole round: the round judged through that version's own launcher and re-checked a digest of its version directory (dedbb877a840...) before every stage, so the CLI could not change mid-round. It is the same Cursor CLI that judged Claude Sonnet 5.5 under v3.23. The Grok model id is the same as in every round; Cursor reports its display name as Grok 4.6 Medium where the frozen settings say Cursor Grok 4.6 Medium, a rename the judging wrapper accepts and recorded 426 times in this round.

The calibration exam

Before scoring a single submission each judge reviews ten held-out programs implementing one ledger specification, from clear to compressed, misleadingly commented, narrated, needlessly abstracted and carrying an embedded instruction to give full marks, five times each in a seeded order. Twenty gates fixed in advance check that the judge sees each construct; a judge may miss at most one gate by at most half a point.

Table 2. Calibration verdicts

Judge	Protocol	Calls	Result	Allowance	Failing gates
Muse Spark 1.3	code-quality-maintenance-v3.24	80	passed	not used	none
Grok 4.6	code-quality-maintenance-v3.24	80	passed	not used	none

Both judges passed every gate with no allowance used. Under v3.23, for Sonnet 5.5, each needed the one-gate allowance.

Table 3. Control means on three of the ten calibration programs

Judge	Control	Naming	Present.	Intent	Struct.	Change.	Verif.
Muse	clear	3.90	3.70	3.50	4.00	4.00	3.50
Muse	compressed	1.00	1.80	1.10	3.70	2.90	3.00
Muse	misleading comments	4.00	3.70	0.30	3.80	2.20	3.20
Grok	clear	3.60	3.50	3.20	4.00	4.00	3.40
Grok	compressed	1.50	1.40	1.50	3.10	2.60	3.00
Grok	misleading comments	3.50	3.30	1.00	3.70	2.20	3.20

All ten controls, every gate value and both judges' control means are in calibration.json.

Results in detail

Table 4. Code quality components at each effort level

Component	Low	Medium	High	Extra-high	Max
Code quality	74.83	76.07	77.60	80.13	81.30
Human readability	71.1	74.1	73.9	76.9	78.7
Maintainability	78.2	79.3	81.0	81.2	82.6
Intent recovery	75.3	74.4	78.0	83.1	83.0
Reviewed score	74.6	76.7	77.4	79.0	80.7
Rated by Muse Spark 1.3	74.1	77.7	78.1	79.4	81.1
Rated by Grok 4.6	75.2	75.7	76.8	78.7	80.3
Standard error of Code quality	1.58	1.55	1.32	1.05	1.03
Hidden behaviours fixed of 231	202	195	215	216	216

Means cover the judged runs per cell (n in Table 1); hidden behaviours cover all 23 runs.

Four runs without a Code quality score

Three safety-classifier stops. On paddockcore at Medium, Extra-high and Max a safety classifier stopped the session while Haiku 5.5 probed the legacy binary, and the run ended with an empty patch, so there is no code to review. As for Claude Opus 5.5's High depotcore, these runs are excluded from judging, count as failed tasks, and are priced and timed. The combined score at those levels is over the judged runs and does not include these failures; tasks passed does. **Max codeccore is unpublished.** Grok 4.6's intent probe gave no valid answer after the single retry: attempt 1 quoted a code line absent from the run, which no recovery rule accepts, and attempt 2 was not valid JSON. The frozen protocol therefore publishes no Code quality score for the run, as for the same task under v3.7 and v3.17. The run passed its tests and counts as passed.

Operator record and calls

Each judge made 80 calibration calls, 112 primary reviews, 5 repeats, 8 pairwise checks and 112 intent probes; Muse made 112 answer-key matches and Grok 111. Muse needed a second attempt on 9 calls and Grok on 5. One Grok review was recovered by the standing recover_excerpts rule: on Extra-high vaultcore both attempts failed only because a quoted line is hard-wrapped in the source, so attempt 1 was selected with the quote re-wrapped to the source's line breaks, scores untouched. Neither judge produced a reviewer fallback. Claude Code's refusal fallback was on and no run used it: every reply came from claude-haiku-5-5.

Cost, tokens and the price tiers

Haiku 5.5 has two list prices per token. Anthropic's pricing page, checked 2026-10-07, charges \$0.10 input, \$0.125 for 5-minute cache writes, \$0.01 for cache reads and \$0.50 output per million tokens for prompts up to 100,000 tokens, and five times as much on every line for longer prompts. Claude Code's receipts hold cumulative session usage, not the size of each request, so each run carries Claude Code's own list-price total (the published cost) beside a floor, every request at the standard tier, and a ceiling, every request at the higher one. Haiku 5.5 ran on a Claude Max subscription, so all three are API-equivalent estimates rather than bills; judging is excluded.

Table 5. Cost with floor and ceiling, raw tokens and runtime by effort

Effort	Runs	\$/task	SE	Floor	Ceiling	Total	Tokens/task	Min/task
Low	23	\$1.04	\$0.33	\$0.24	\$1.20	\$23.82	14.51M	22.9
Medium	23	\$1.36	\$0.52	\$0.31	\$1.53	\$31.17	19.47M	24.0
High	23	\$1.25	\$0.34	\$0.28	\$1.40	\$28.69	16.98M	26.8
Extra-high	23	\$0.84	\$0.16	\$0.20	\$0.98	\$19.31	9.34M	19.0
Max	23	\$1.54	\$0.28	\$0.33	\$1.65	\$35.33	17.97M	27.7
Full sweep	115	\$1.20		\$0.27	\$1.35	\$138.31	1,800M	46.2 h

Limitations recorded with the estimates

- Haiku 5.5 is priced by prompt length: prompts over 100,000 tokens cost five times the standard rate on every line. Claude Code receipts hold cumulative session usage, not per-request prompt sizes, so a run's true API price cannot be split between the tiers. Each run carries Claude Code's own list-price total, which prices each request at its own tier, beside a floor (every request at the standard tier) and a ceiling (every request at the over-100k tier).
- Floor and ceiling price cache writes at the 5-minute rate, as the harness cards do. Two runs (medium lodgecore and max depotcore) wrote cache entries with the 1-hour TTL, which costs more, so Claude Code's total for them sits up to 0.4% above the ceiling.
- Cost per task covers all 23 runs in a cell. The harness cards average the judged runs, so their figures at medium, extra-high and max (\$1.41, \$0.87 and \$1.55) differ slightly from this bundle's (\$1.36, \$0.84 and \$1.54).
- No Batch, Fast or priority pricing is applied.

Beside Claude Sonnet 5.5 and GPT-6.1 Sol

Sonnet 5.5 and GPT-6.1 Sol ran the same 23 tasks, judged by the same judges under v3.23 and v3.18; their numbers are the published rows and nothing was re-judged. The site's comparison page adds GPT-6 Luna. Harnesses (Claude Code 2.1.293 to 2.1.296, Claude Code 2.1.291 to 2.1.293, Codex 0.159.0) and cost bases (Claude Code's own total for the Claude columns, API-equivalent from the token ledger for GPT-6.1 Sol) differ, so small gaps are confounded.

Table 6. Haiku 5.5, Sonnet 5.5 and GPT-6.1 Sol by effort

Effort	Combined	Passed of 23	Code quality	\$/task	Min/task
Low	84.42 / 81.63 / 86.22	13 / 15 / 20	74.8 / 61.7 / 70.8	\$1.04 / \$2.88 / \$0.31	22.9 / 20.0 / 6.9
Medium	84.03 / 84.10 / 86.44	13 / 17 / 22	76.1 / 65.4 / 70.2	\$1.36 / \$3.00 / \$0.40	24.0 / 20.8 / 10.6
High	88.01 / 86.79 / 88.23	17 / 20 / 23	77.6 / 70.8 / 73.7	\$1.25 / \$2.39 / \$0.33	26.8 / 14.8 / 10.2
Extra-high	91.83 / 90.20 / 88.78	22 / 22 / 23	80.1 / 75.8 / 75.0	\$0.84 / \$3.38 / \$0.43	19.0 / 21.6 / 15.4
Max	92.16 / 92.02 / 88.35	22 / 23 / 23	81.3 / 81.7 / 73.8	\$1.54 / \$5.52 / \$0.46	27.7 / 38.0 / 14.5

Each cell reads Haiku 5.5 / Sonnet 5.5 / GPT-6.1 Sol. Combined scores and Code quality are over judged runs; pass counts, cost and minutes cover all 23 runs per cell.

At Max Haiku 5.5 and Sonnet 5.5 are effectively tied (92.16 against 92.02, a 0.14-point gap far inside the error), with Haiku 3.6 times cheaper per task there. Below Max Haiku rates higher on Code quality but passes fewer tasks than Sonnet 5.5 from Low to High. GPT-6.1 Sol solves more tasks than Haiku 5.5 at every level for less money, and Haiku scores about 3 points higher at Extra-high and Max.

On the Frontier v4 board

Haiku 5.5's best level, Max at 92.16 (SE 0.37, n=21), ranks 4 of the board's 63 columns, behind only Grok 4.7's top three levels, which are judged by a different panel (Muse Spark 1.3 and GPT-6.1 Sol). Among columns judged by Muse Spark 1.3 and Grok 4.6 it is the highest point estimate, 0.14 above Sonnet 5.5 at Max (92.02, SE 0.35), well within the errors. Its levels rank 37, 40, 25, 7, 4 from Low to Max.

How it was run

From a tagged run worktree. The sweep ran from bench/2026-10-07-haiku55-frontier-v4, so every run summary carries a clean source block and records the frozen suite lock's task hash; no hash bridge was needed. **Claude Code versions.** Low ran 22 tasks on 2.1.293 and 1 on 2.1.294; Medium 3 on 2.1.294 and 20 on 2.1.295; High on 2.1.295; Extra-high 2 on 2.1.295 and 21 on 2.1.296; Max on 2.1.296. **Two pauses between tasks.** The sweep was stopped from October 9, 15:43 UTC, to October 10, 09:20 UTC, and from October 10, 19:48 UTC, to October 11, 15:50 UTC. Each time the run in progress was set aside and re-run, and no counted run was in progress across a pause, so run timings are intact. The host's clock moved from PDT to CEST during the sweep; times are UTC. **Where and how.** --billing subscription on a Claude Max plan, the local sandbox on the owner's Mac, a flat 3-hour task timeout, one task at a time, from October 8, 10:40 UTC, to October 11, 22:49 UTC. Judging ran October 11, 22:52 UTC, to October 12, 10:11 UTC, with no sweep running. Anthropic's default effort for Haiku 5.5 is Medium on the API and in Claude Code. **One attempt per task and level;** effort labels are Claude Code's own.

Evidence and reproduction

Public files: `runs.json` and `runs.csv` with every run's factors, hidden behaviours fixed, integrity verdicts, task hash, Claude Code version, replies by serving model, cost with its tier floor and ceiling, and each judge's six-dimension sub-scores and intent recovery; `groups.json`; `calibration.json` with every gate value and control mean; `judge-protocols.json` with the exact rubric, system text, probe and match instructions, schemas, weights and the judge settings and Cursor version record; `economics.json` with cost and token aggregates, both price tiers and pricing limitations; and `provenance.json` with source hashes and limits.

<https://github.com/morganlinton/VulcanBenchCOM/tree/main/assets/data/swe-v4-haiku55-v324>

Withheld: raw prompts and responses; submitted patches and reconstructed sources; quirk answer keys (they describe hidden-test behaviour); reviewer session identifiers and usage receipts; the unused Grok probe attempts on max codecore; host binary paths.

Recompute the published numbers

Each judged run's Code quality is $(0.24 \times \text{reviewed score} + 0.09 \times \text{intent recovery}) / 0.33$, where both terms are the equal mean of the two judges. The combined score is $100 \times (0.50 F + 0.085 Q + 0.085 S + 0.33 C / 100)$ with F, Q, S on a 0 to 1 scale. Group means weight the cell's judged tasks equally. The export script recomputes every row from the frozen summary, checks every run's cost against Claude Code's stream, every task hash against the suite lock, the recovered protocols and the Cursor version pin, and the harness card tables, and refuses to write if any value differs.

What this report does not claim

No human rated anything; the scores are model judgment for a human reader, not human validation. Standard errors describe task sampling only. The measured-maintenance layer is unbuilt, so the 33 points are 24 reviewed plus 9 intent recovery. Runs were not repeated. Hash checks bind the export to frozen files; they do not prove that judges were unbiased or that no training overlap exists.

Appendix. Per task at Low, Medium and Max

Task	Low CQ	Max CQ	Low combined	Medium combined	Max combined
blendcore	68.6	88.0	88.25	89.18	93.90
cellarcore	67.8	80.4	87.42	75.57	91.58
codeccore	82.1	n/a	67.60	74.12	n/a
depotcore	69.4	77.0	80.67	84.75	90.26
freightcore	85.6	88.6	93.81	91.06	94.86
granarycore	63.1	68.0	81.62	85.39	86.89
hedgcore	80.4	85.5	92.15	94.00	93.74
lodgcore	76.5	85.1	69.36	81.44	93.01
matchcore	76.5	82.6	90.57	92.66	92.64
metercore	81.2	80.4	92.42	91.36	92.14
pacecore	73.4	87.1	89.72	43.70	94.26
paddockcore	53.4	n/a	58.83	n/a	n/a
payrollcore	74.2	77.3	84.81	88.70	90.77
qlite	84.1	79.5	93.20	88.23	91.78
queuecore	76.5	79.5	78.31	77.91	91.81
quotacore	82.6	83.3	92.97	92.84	93.16
reflowcore	67.2	75.8	87.89	89.36	90.71
schedcore	79.0	82.5	85.22	89.19	92.55
settlecore	71.9	84.1	74.00	92.51	93.23
snapcore	74.2	79.5	90.16	64.39	91.80
stampcore	78.1	79.5	91.31	90.68	91.82
tallycore	81.4	79.9	92.49	91.05	91.98
vaultcore	73.9	83.5	78.97	80.61	92.50

Per-task values from runs.json. Task names are shortened for width. n/a: no Code quality score (classifier stop or unpublished probe).

Source hashes

Frozen artifact	SHA-256
v3.24/summary.json	3c57f31901d4a55b87e131727c301132...
v3.24/protocol.json	15c04683b3056999bec21cd99b148fc4...
v3.24/private-manifest.json	7b4ca5eb516540ad30a86a0015b18a2a...
v3.24/calibration-muse.json	203551085fe21fe627cb154d17f6c336...
v3.24/calibration-grok.json	5f5a3010263cb0c420ae2a54fa0e6628...
calls/grok/probe/submission-030/operator-invalid.json	d582657e4d5f0a339c991c6c1faaec19...
comparison.json	545c448b6f1cfb3ea0f680121081178a...
suite.lock.json	861da9e5386e88e6c77236ee5326863e...
haiku55-v324-efforts.csv	672654a2dc979a7ebc11fd9b3b83a322...
haiku55-vs-sonnet55-efforts.csv	e370f98d7b4fc74ca01cca295604ce4b...
haiku55-vs-gpt61sol-efforts.csv	ffb26c90a4b1c7a6cf90817db28040bb...
haiku55-vs-luna-efforts.csv	ffd1b79c41903764a0455647ba41ce90...

VulcanBench Frontier v4: Claude Haiku 5.5 across effort levels

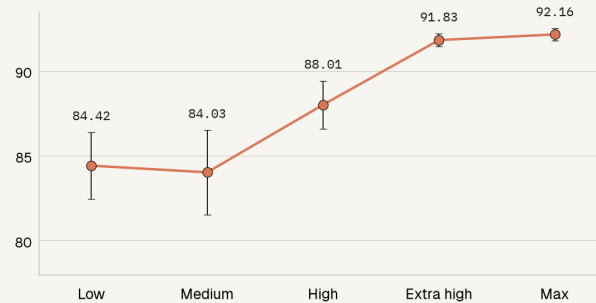
Combined score and runtime at every effort level, 23 tasks per effort. Code quality judged by Muse Spark 1.3 and Grok 4.6.

● **Claude Haiku 5.5** · **Claude Code 2.1.293 to 2.1.296**

n=23 at every effort, n=22 judged at medium, n=22 judged at extra high, n=21 judged at max

Combined score

/100 · higher is better · focused scale



Mean runtime

Minutes per task · lower is better

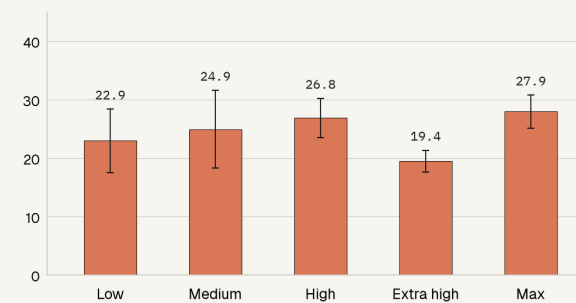


Table 1 | Code quality at each effort level

Component	Low	Medium	High	Extra high	Max
/100, mean of both judges					
Code quality	74.83	76.07	77.60	80.13	81.30
Human readability	71.1	74.1	73.9	76.9	78.7
Maintainability	78.2	79.3	81.0	81.2	82.6
Intent recovery	75.3	74.4	78.0	83.1	83.0
Rated by Muse Spark 1.3	74.1	77.7	78.1	79.4	81.1
Rated by Grok 4.6	75.2	75.7	76.8	78.7	80.3
Standard error of Code quality	1.58	1.55	1.32	1.05	1.03
Tasks passed of 23 (every run)	13	13	17	22	22
Runs with fallback-model turns	0	0	0	0	0
Cost per task, USD (Claude Code total)	1.04	1.41	1.25	0.87	1.55
Floor, every prompt under 100k	0.24	0.32	0.28	0.20	0.33
Ceiling, every prompt over 100k	1.20	1.59	1.40	1.01	1.66

Claude Code 2.1.293 for low (22 runs); 2.1.294 for low (1 run), medium (3 runs); 2.1.295 for medium (20 runs), high, extra high (2 runs); 2.1.296 for extra high (21 runs), max. Earlier Claude columns used older CLIs. Refusal fallback on; no run used it. Paddockcore at medium, extra high and max is not judged: a safety classifier stopped the session while the model probed the legacy binary and it ended without editing the module, so the run counts as failed. Max codecore has no published Code quality score: a judge's intent probe had no valid response on either attempt, so the frozen protocol leaves it unpublished.

Haiku 5.5 costs five times as much for prompts over 100,000 tokens. Run receipts hold session totals, so the floor prices every request at the standard tier and the ceiling at the higher one; Claude Code's own total sits between them. The sweep paused twice between tasks (October 9 to 10 and October 10 to 11), which leaves every run's timing intact.

Judge settings are the original v3.3 and v3.4 protocols. Muse ran the same binary (sha256 match); Grok ran through Cursor CLI 202610.01, pinned by version for the whole round, the same CLI that judged Sonnet 5.5. Calibration: both judges passed every gate. Frontier Code quality is L1 plus L2 and is not comparable with Routine v1 Code quality.



VulcanBench Frontier v4: Claude Haiku 5.5 and Claude Sonnet 5.5

Best combined score per model: Claude Haiku 5.5 92.16 at max · Claude Sonnet 5.5 92.02 at max.

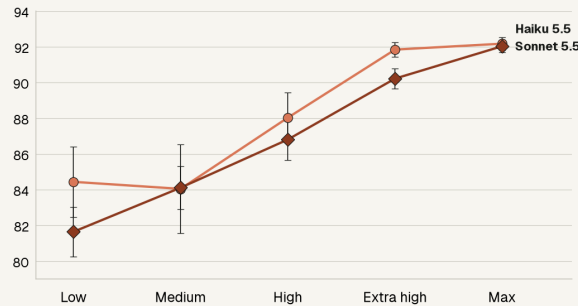
23 hard legacy-reconstruction tasks per effort level. Code quality judged by Muse Spark 1.3 and Grok 4.6.

● Claude Haiku 5.5 (with fallback)

◆ Claude Sonnet 5.5 (with fallback)

Combined score

/100 · higher is better · focused scale · whiskers +/-1 se



Cost per task

USD, API-equivalent at list prices · every serving model included · bases in notes

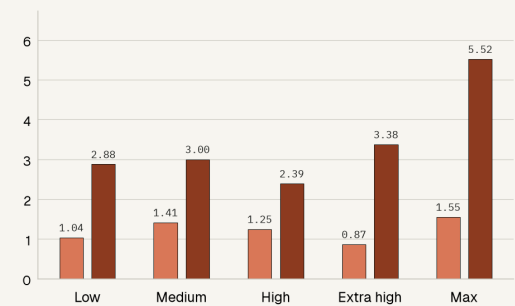


Table 1 | Every effort level

Model and measure	Low	Medium	High	Extra high	Max
● Claude Haiku 5.5					
Combined score	84.42	84.03*	88.01	91.83*	92.16*
Tasks passed of 23	13	13	17	22	22
Cost per task, USD	1.04	1.41	1.25	0.87	1.55
Fallback share of replies (runs)	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)
◆ Claude Sonnet 5.5					
Combined score	81.63	84.10	86.79	90.20	92.02
Tasks passed of 23	15	17	20	22	23
Cost per task, USD	2.88	3.00	2.39	3.38	5.52
Fallback share of replies (runs)	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)

* Combined score over the judged runs when fewer than 23. Bold marks each model's best level. Tasks passed counts all 23 runs.

Protocols: Haiku 5.5 is Code quality v3.24 and Claude Sonnet 5.5 is v3.23. Same rubric, controls, weights and judges (Muse Spark 1.3 and Grok 4.6, neutral for both labs), judged in separate sessions. Unjudged Haiku runs: medium paddockcore, extra high paddockcore, max paddockcore (a safety classifier stopped the session before any edit; counted as failed), and max codeccore (no valid judge probe, so unpublished under the frozen protocol).

Both ran through Claude Code with the refusal fallback on, and neither used it; costs are Claude Code's own list-price totals (Haiku 5.5 sits between the floor and ceiling of its two price tiers, shown on its own card). Claude Code versions differ (21.293 to 21.296 against 21.291 to 21.293), and both rounds' Grok calls ran on Cursor CLI 202610.01.