Frontier Coding Accuracy Has Converged. Efficiency Has Not.

An evaluation of Claude Fable 5 (minimum reasoning effort), Claude Sonnet 5 (maximum effort), and GLM-5.2 on ten decontaminated tasks from real open-source repositories

VulcanBench (github.com/morganlinton/VulcanBench)

ABSTRACT

We evaluate three contemporary large language models as autonomous coding agents on VulcanBench v2, a ten-task suite built exclusively from real merged pull requests in production open-source repositories (flask, aiohttp, sqlglot, click, chi, more-itertools, packaging), each merged after the models' training cutoffs and graded by deterministic hidden tests. All three models achieve the same 80% pass rate and fail the same two tasks. At equal accuracy, resource consumption diverges by nearly an order of magnitude: Claude Fable 5 running at its lowest reasoning effort completes the suite for $2.27 in 14 minutes using 0.14M tokens, versus $8.41 / 59 minutes / 1.75M tokens for Claude Sonnet 5 at its highest effort, and $15.81 / 124 minutes / 10.82M tokens for GLM-5.2. We additionally document a safety-classifier refusal by Claude Fable 5 on a benign HTTP-parser bug fix, an operational failure mode absent from conventional leaderboards. We conclude that on ticket-sized engineering work, correctness has ceased to discriminate frontier models; cost, latency, and availability now carry the signal.

1. Introduction

Public coding benchmarks saturate quickly: when frontier models score 98-100%, a benchmark measures nothing but its own ceiling. VulcanBench v2 was constructed to restore discrimination by sourcing tasks from real engineering work that current models demonstrably cannot all solve, and by treating economics (cost, wall-clock time, tokens, steps) as first-class measurements rather than footnotes.

This report asks a narrow question with a practical payoff: does the newest frontier tier at its cheapest setting match the previous tier at its most expensive? Anthropic's documentation claims Claude Fable 5 at low effort often exceeds prior models at maximum effort. We test that claim on tasks that cannot be answered from memorization, and add GLM-5.2 as a budget-tier reference point.

2. Benchmark Design

2.1 Task sourcing and decontamination

Every task originates from a real pull request merged into a production open-source repository between February and June 2026, after the training cutoffs of all evaluated models, so no model has seen the fix. The agent receives the repository sliced at the PR's base commit plus a terse issue describing the symptom (never the fix), mirroring how an engineering team assigns a ticket. Tasks span two languages (Python, Go), seven repositories, and two categories (bug fix, feature).

2.2 Grading

Grading is fully deterministic: hidden tests, authored against the upstream project's public API and validated to fail on the base commit and pass on the maintainer's merged fix (three consecutive validation runs each), decide pass or fail. No LLM judges are used. Each task also carries a regression test that must keep passing. Agents run

in a network-isolated Docker sandbox; tasks with heavy dependency stacks use per-task images with the upstream package's dependencies preinstalled.

2.3 Suite composition

The ten tasks were composed from a validated pool of seventeen so that measured difficulty spans the range from routine (single-site library fixes) to unsolved (a multi-site behavioral redesign in Flask that no evaluated model has ever passed). Composition targets each frontier model landing in the 70-90% band, keeping every task's outcome informative.

Table 1. The ten tasks. Each is a real merged PR; the hidden test grades the stated requirement.

Task	Upstream PR	What the agent must do
flask-teardown-robust	flask#5928	Robust teardown: all callbacks run, errors aggregate
aiohttp-upgrade-deferred	aiohttp#13016	Defer protocol upgrade until request body is read
sqlglot-qualify-lateral-star	sqlglot#7802	Expand SELECT * through CROSS JOIN LATERAL
sqlglot-iso8601-nanos	sqlglot#7808	Transpile FROM_ISO8601_TIMESTAMP_NANOS per dialect
sqlglot-parser-tuple-subquery	sqlglot#7807	Parser: tuple vs parenthesized expression
click-version-distribution-name	click#3582	Resolve module name to installed distribution
chi-readfrom-tee-doublecount	chi#1085 (Go)	ReadFrom byte double-count with Tee writer
more-itertools-interleave-empty	more-itertools#1193	interleave_evenly on empty input
packaging-range-prerelease-policy	packaging#1295	VersionRange union prerelease policy
packaging-empty-name	packaging#1305	Reject wheel filename with empty project name

3. Experimental Setup

Each model ran the full suite once (one attempt per task) through the same agentic harness: identical tools (shell, file read/write/edit), identical per-task step budgets, a 30-minute per-task wall-clock limit, and Anthropic prompt caching enabled where supported. Claude Fable 5 ran at effort=low; Claude Sonnet 5 at effort=high; GLM-5.2 was requested at effort=high, but its API exposes no reasoning-effort control, so it ran at the provider default (recorded as metadata only). Claude Fable 5 and Claude Sonnet 5 share a tokenizer, making their token counts directly comparable. No refusal fallbacks were configured: a safety-classifier refusal is recorded as a failed task, since an engineering team experiences it as one. Costs are computed from provider list prices: $10/$50 per million input/output tokens (Fable 5), $3/$15 (Sonnet 5), $1.40/$4.40 (GLM-5.2).

4. Results

Table 2. Aggregate results. Fable 5 attempted 9 of 10 tasks; the aiohttp task was refused by its safety classifier and is scored as a failure. $/solved = total cost divided by tasks passed.

Model (effort)	Score	Cost	Time	Tokens	Steps	$/solved
Claude Fable 5 (low)	8/10	$2.27	14 min	0.14 M	490	$0.28
Claude Sonnet 5 (high)	8/10	$8.41	59 min	1.75 M	1,762	$1.05
GLM-5.2 (high)	8/10	$15.81	124 min	10.82 M	2,698	$1.98

Figure 1. Resource consumption at equal (80%) accuracy. Within each panel, bars are Fable 5 (dark blue), Sonnet 5 (slate), GLM-5.2 (rust).

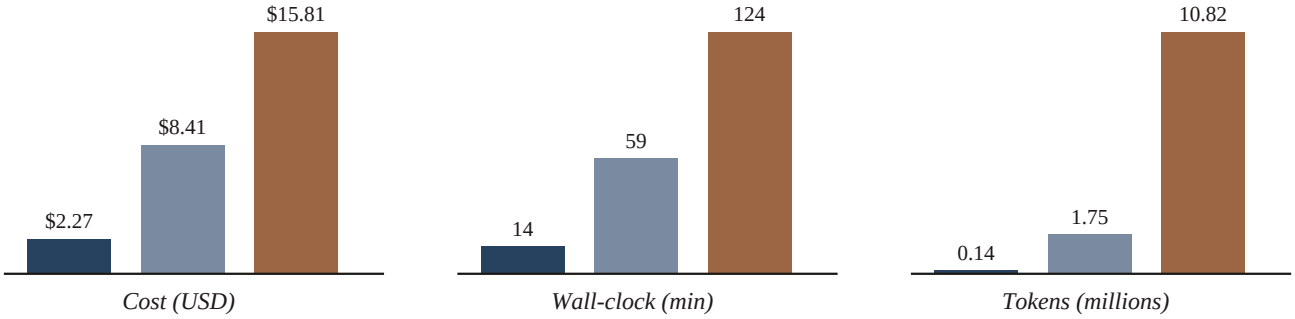

Table 3. Per-task outcomes (pass / fail, cost, agent steps). REFUSED: the model's safety classifier declined the task before the first response.

Task	Fable 5 (low)	Sonnet 5 (high)	GLM-5.2
flask-teardown-robust	FAIL $0.38 / 62	FAIL $1.15 / 314	FAIL $1.98 / 586
aiohttp-upgrade-deferred	REFUSED	FAIL $4.26 / 370	FAIL $2.66 / 256
sqlglot-qualify-lateral-star	pass $0.57 / 86	pass $1.52 / 330	pass $3.86 / 396
sqlglot-iso8601-nanos	pass $0.34 / 60	pass $0.52 / 204	pass $2.54 / 344
sqlglot-parser-tuple-subquery	pass $0.16 / 46	pass $0.11 / 86	pass $1.10 / 254
click-version-distribution-name	pass $0.18 / 46	pass $0.19 / 122	pass $1.02 / 262
chi-readfrom-tee-doublecount	pass $0.14 / 56	pass $0.18 / 86	pass $0.48 / 128
more-itertools-interleave-empty	pass $0.09 / 42	pass $0.05 / 54	pass $0.10 / 98
packaging-range-prerelease-policy	pass $0.23 / 48	pass $0.37 / 142	pass $1.80 / 254
packaging-empty-name	pass $0.18 / 44	pass $0.08 / 54	pass $0.28 / 120

4.1 Accuracy converged

All three models pass exactly eight of ten tasks, and the two failures are the same tasks for every model. The Flask teardown redesign (flask#5928) requires making every teardown callback and signal receiver run despite earlier exceptions, with errors aggregated into nested exception groups. Every model fixes one call site and misses the siblings: an incomplete multi-site fix, each time. The aiohttp deferred-upgrade task fails for Sonnet 5 and GLM-5.2 on the protocol state machine itself, and was refused outright by Fable 5.

4.2 Minimum effort matched maximum effort, at a fraction of the resources

Fable 5 at effort=low used 490 agent steps across the suite where Sonnet 5 at effort=high used 1,762 and GLM-5.2 used 2,698. It solved qualify-lateral-star, historically the suite's strongest discriminator (Sonnet 5 lifetime record: 1 pass in 6 attempts), in 86 steps for $0.57. Per solved task, Fable 5 cost $0.28 versus $1.05 (Sonnet 5) and $1.98 (GLM-5.2).

4.3 The refusal

On the aiohttp task, Fable 5's cyber-safety classifier declined the request before the first token (category 'cyber'). The task is a benign bug fix: an HTTP parser must defer a websocket protocol switch until the request body is fully read (RFC 9110 section 7.8). The refusal is a documented false-positive mode for security-adjacent content, but it is invisible on conventional benchmarks and highly visible to an engineering team whose model declines to touch parser code. We deliberately configured no fallback model, so the refusal scores as a failure.

4.4 Cheap tokens, expensive outcomes

GLM-5.2 has the lowest list price of the three, yet produced the most expensive run: 10.82M tokens (77x Fable 5), 124 minutes of wall-clock time, and the highest cost per solved task. On agentic work, token efficiency dominates unit price.

5. Discussion

The result pattern inverts the usual benchmark question. When three models spanning a 7x price range and two vendors land on identical pass rates with identical failure sets, per-task correctness has stopped carrying information about model choice; what differs is what each model spends to get there. For teams selecting a model for ticket-sized engineering work, the operative questions are cost per solved task, latency, and operational availability (including refusal behavior on security-adjacent code), not leaderboard accuracy.

The Flask task deserves note: it has now been attempted by four frontier models (including Claude Opus 4.8 in earlier measurement, 0 for 4 across attempts) and solved by none. Its difficulty shape, a behavioral contract that must be implemented consistently across several call sites and is graded on complete structural parity, appears to be a genuine frontier capability gap rather than a quirk of any one model.

6. Limitations

n=1 per cell: single runs per task per model. Earlier repeated measurements on this suite show run-to-run variance on the hardest tasks (Sonnet 5 passed lateral-star in this run after failing it in five prior attempts; GLM-5.2 also passed it here), so per-task comparisons should be read as indicative, not definitive. Aggregate economics are more stable: the cost and token gaps exceed plausible variance by an order of magnitude. GLM-5.2's effort setting is not honored by its API; its leg also required a budget-cap restart, completed on the same day with identical configuration. The suite measures ticket-sized, issue-driven work (patches under roughly 200 lines); it does not measure multi-day projects, greenfield design, or requirement negotiation. Ten tasks is a discrimination instrument, not a labor-market survey.

7. Reproducibility

VulcanBench is open source (github.com/morganlinton/VulcanBench). Tasks, hidden tests, gold patches, per-task Docker images, and the harness (including prompt caching and deterministic verification) are in the repository; every task validates as gold=1.0, pre-patch=0.0, deterministic over three runs. A full two-model measurement of this suite costs roughly $9-16 at current list prices. Raw run artifacts for this report (traces, patches, per-run summaries) total 30 runs across the three models.