Muse Spark 1.2 *(low vs. high vs. xhigh effort)*

Twenty-three real merged PRs across Python, Rust, TypeScript, JavaScript, and Go, graded by deterministic hidden tests

The steepest backward effort knob measured on v3: 87.0% at low, 52.2% at xhigh. Raising reasoning effort converts finished wrong answers into wall-clock timeouts (0 → 5 → 10) under the same 20/45/60-minute budgets every model on the board gets. Ten of the fifteen timed-out runs had written no patch at all: the model reasoned for its entire budget and never touched the repo. At low effort it ties the board's 87.0% cluster at $0.41 per solved task.

Table 1. All three configurations, one attempt per task. Wrong = finished run that failed the hidden tests; timeout = cut off at its 20/45/60-minute budget.

Effort	pass@1	Solved	Wrong	Timeouts	Cost	Tokens/task	Time/task	$/solved
low	**87.0%**	**20/23**	**3**	**0**	**$8.19**	**206 K**	**6.7 min**	**$0.41**
high	73.9%	17/23	1	5	$20.92	483 K	21.3 min	$1.23
xhigh	52.2%	12/23	1	10	$27.25	658 K	28.7 min	$2.27

1. The mechanism is slower steps, not harder work. At higher effort the model emits 2 to 33× the output tokens per agent step on the same task (*itertools-strip-prefix*: ~140 tokens/step, solved in 7 minutes at low; ~4,600 tokens/step, killed at the 60-minute wall at xhigh). All 15 timeouts ran out the wall clock; none hit the step ceiling or a cost cap. Partial work is graded at kill time; 10 of 15 timed-out patches were zero bytes.

2. The regression sits in solvable work. Eight of the fifteen timed-out cells are tasks the same model solves at low in 1.6 to 12 minutes. Three tasks (*networkx-leiden*, *pennylane-trotter*, *sqlglot-canonicalize*) score zero at every setting; extra reasoning rescues none of them.

3. Fourth model, same shape. Qwen3.8-Max fell 26.1 points across its knob (No. 12), GLM 5.3's raw API 13.1 (No. 18), and Claude Opus 5 fell 8.7 (No. 10). Muse Spark's 34.8 is the steepest, and its low column is the strongest debut of the four.

Table 2. The eleven tasks that moved; the other twelve scored identically at every setting.

Task	low	high	xhigh
aiohttp-upgrade-deferred	*solved*	*timeout*	*timeout*
flask-teardown-robust	*solved*	*timeout*	*timeout*
itertools-strip-prefix	*solved*	*solved*	*timeout*
jiff-strftime-negpad	*solved*	*solved*	*timeout*
packaging-range-prerelease-policy	*solved*	*solved*	*timeout*
semver-inc-dotted-prerelease	*solved*	*solved*	*timeout*
semver-xrange-order	*solved*	*solved*	*timeout*
sqlglot-qualify-lateral-star	*solved*	*wrong*	*wrong*
networkx-leiden-communities	*wrong*	*timeout*	*timeout*
pennylane-trotter-fragmented	*wrong*	*timeout*	*timeout*
sqlglot-canonicalize-internal-names	*wrong*	*timeout*	*timeout*

Reading the map. Low fails by answering wrong; high and xhigh fail by running out the clock. No task anywhere in the matrix is solved at xhigh and missed at low.

Caveats. Single pass per task (±7 to 10 points stderr per column), so high vs. xhigh ordering is suggestive while the 34.8-point inversion and the timeout composition are categorical. *minimal* and *medium* are untested, and Meta does not document the unset default, so this report cannot say what an untuned integration gets.

Method. VulcanBench v3: 23 tasks from real merged post-cutoff PRs, network-isolated Docker sandboxes, deterministic hidden-test grading, one attempt per task per configuration. Wall-clock budgets 20/45/60 min by repo size, identical for every model, unchanged since Reports No. 07 to 10. Muse Spark 1.2 at $1.25/$4.25 per M tokens via the Meta Model API; $56.36 across 69 runs; 0 contaminated runs; effort verified per run. Full model card and per-task matrix: github.com/morganlinton/VulcanBench.