GLM 5.3 in ZCode vs. a bare-bones harness

One model, two delivery systems: Z.ai's own ZCode harness on a GLM Coding Plan, and the metered z.ai API through a minimal reference loop, graded by the same hidden tests.

Same model, 21.8 points apart: 65.2% through the raw API at max effort, 87.0% inside ZCode. The effort knob points opposite directions: raw, it inverts (78.3 to 73.9 to 65.2 as effort rises); inside ZCode it holds then climbs (82.6, 82.6, 87.0). Even ZCode's worst column beats the raw API's best. The gap is unfinished work, not worse work: 18 of the raw loop's 19 failures are wall-clock timeouts, while ZCode never times out and every one of its 11 failures is a finished wrong answer. On the v3 best-effort board the raw-API entry lands 12th of 14; the same model inside its own product scores with the 87% frontier cluster.

Results (one attempt per cell, ±1 stderr)

Effort	In ZCode	Bare-bones API	Delta	ZCode time/task	API time/task
low	82.6% ±7.9	78.3% ±8.6	+4.3	**4.8 min**	8.4 min
high	82.6% ±7.9	73.9% ±9.2	+8.7	**4.6 min**	13.5 min
max	**87.0% ±7.0**	65.2% ±9.9	**+21.8**	**5.3 min**	22.8 min

Findings

1. Worth 21.8 points at the shipped default. Max is GLM 5.3's default effort: an untuned API integration gets 65.2%, Z.ai's own product 87.0%, from identical weights. No effort setting lets the bare loop catch the product.

2. Opposite knob directions. Raw runs backward (the Qwen and Grok inversion of Reports 12 and 14); ZCode holds then rises. Spread across the knob: 4.4 points in ZCode, 13.1 raw. The harness matters more than the effort.

3. Failure modes split perfectly. Raw: 18 timeouts, 1 wrong answer, 8.4 to 22.8 min/task as effort rises. ZCode: 0 timeouts, 11 wrong answers, ~5 min/task flat. The product drives every run to a finished answer.

4. Three tasks the raw API never finishes, ZCode completes every time, solving two (pennylane-trotter-fragmented falls at max, a task the raw API and Qwen3.8-Max never finished). sqlglot-canonicalize remains Grok Build's alone (Report 16).

5. Receipts. All 69 ZCode runs verified executing glm-5.3 at the requested thought level from its session store, moving ~5x the tokens of the bare loop. Raw sweep $35.48 metered; ZCode $0 marginal cash on the Coding Plan (~$87 API-equivalent). 138/138 integrity audits clean.

The ten tasks that moved (S solved · W wrong · T timeout)

Task	API l/h/m	ZCode l/h/m
pennylane-trotter-fragmented	T T T	W W **S**
networkx-leiden-communities	T T T	**S S** W
sqlglot-canonicalize-internal-names	T T T	W W W
sqlglot-iso8601-nanos	T T **S**	**S S S**
aiohttp-upgrade-deferred	**S S** T	**S S S**
flask-teardown-robust	**S** T T	**S** W W
jiff-strftime-negpad	T **S S**	W W **S**
semver-inc-dotted-prerelease	**S S** T	**S S S**
semver-xrange-order	**S S** T	**S S S**
semver-truncate	**S S** W	W **S S**

The other thirteen tasks were solved by both systems at every level. Where the raw API degrades it degrades into T; where ZCode degrades it degrades into W.

Method. VulcanBench v3: 23 real merged post-cutoff PRs (Python 9, TS 4, Rust 4, Go 3, JS 3), network-isolated Docker verification, deterministic hidden tests, judges off. Both systems ran 2026-08-22, one attempt per task per level. ZCode driven headlessly (zcode-app-cli 3.8.1-15, runtime 0.16.3) with web tools removed, memory off, workspace outside the repository; thought level pinned per run and read back as proof. Wall-clock budgets 45/60 min by repo size. Full study: vulcanbench.com/benchmarks/18-glm53-zcode-harness.html