Brute Force vs. Finesse: How Chinese-Flagship and Western LLMs Solve Middle-School Competition Math

A cost-matched, difficulty-resolved benchmark of 16 models on AMC 8 / AJHSME

GenAware Model Lab · 31 May 2026

Executive summary

> **Governing thought.** On hard reasoning, **capability has commoditised but speed has not — and speed is where the revenue is.** Eight Chinese flagships now match the Western frontier (Claude Opus 4.8, Sonnet 4.6) on accuracy, even on the hardest problems; what still separates them is latency, and latency is precisely the axis that gates the revenue-dense, interactive end of the market. The strategic prize is the **fast-and-accurate** quadrant — today held by the frontier alone, and credibly contestable by exactly one challenger: **DeepSeek**.

We benchmarked **16 leading models** on **4,896 graded attempts** of middle-school competition math (AMC 8 / AJHSME), and — the decisive design choice — resolved every result by **problem difficulty** rather than reporting one average. That choice is the whole game: pooled, fourteen of sixteen models huddle between 84% and 99% and look interchangeable; split by difficulty, they fall into a clean strategic map. Three findings, each a function of difficulty:

1. **Capability parity is real — but only visible on hard problems.** On easy questions everyone scores 92–100%. On the hardest tier the field splits: the eight Chinese flagships and the two Western frontier models **hold** at 95–100%, while the cost-matched Western budget models **collapse to ≈66–81%**. On accuracy, the Chinese flagships have genuinely caught the frontier.
2. **The frontier's moat is speed, not accuracy.** The frontier is difficulty-inelastic — 1–3 s at every tier — while the accurate Chinese models climb to **12–34 s on hard problems**. Latency is a product gate, not a dial, and it compounds in agentic loops (a 20-step task runs ≈40 s at frontier speed vs. **5–7 minutes** on a flagship). The frontier leads on the axis that is hardest to copy.
3. **Usage is not revenue.** Cost-per-correct — where DeepSeek (0.077 ¢) and the budget models lead — measures the high-volume, low-margin usage lane. Revenue is Pareto-distributed and sits at the **low-latency × high-accuracy** corner, which only the frontier fills. Among serious buyers the binding constraint is **rate-limit capacity, not sticker price** — and the Chinese flagships' token-burn makes that constraint worse.

The positioning in one exhibit — speed against accuracy-on-hard:

Accuracy on hard ↓ / Speed →	Slow (≥ 8 s)	Fast (< 3 s)
Holds (≥ 95%)	**Usage lane** — DeepSeek, Qwen, GLM, Kimi, Seed, Step (Chinese flagships). Cheap per correct, but locked out of interactive products. Wins volume, not margin.	**Revenue lane (the prize)** — Opus, Sonnet. Fast and right when it's hard — the only cell that monetises at a premium.
Cracks (< 90%)	**Dead lane** — ERNIE, MiniMax, Llama. Slow and unreliable on hard problems. No reason to deploy.	**Commodity front-end** — Haiku, GPT-mini, GPT-nano, Gemini, Grok. Fine on easy volume; crack on hard — need an escalation path to a stronger model.

The prize is the top-right. DeepSeek sits one axis away — frontier accuracy at commodity cost, held back only by latency — so its strategic trajectory is **leftward, into the revenue quadrant**,

conditional on three gates: sustainable profitability at its price points, capacity to serve volume, and closing the latency gap (§10).

Where to play.

- **Balanced default: Sonnet 4.6** — fast, cheap, 95%; the cutoff every other model must clear.
- **Easy, high-volume work:** the cheapest competent model (Gemini Flash-Lite, GPT-nano, or DeepSeek) — accuracy is a non-issue; pay the least.
- **Hard or interactive work (where the revenue is): Opus 4.8** — the only model both fast and accurate when it counts.
- **Cost-sensitive, latency-tolerant batch: DeepSeek** — a no-regrets pick today, and the one model to watch as a future threat to the premium tier.
- **Avoid for production:** the verbose flagships (Kimi, Qwen) — too slow for the revenue lane, and dearer per correct answer than Sonnet on hard problems.

---

1. The buyer's question

Chinese frontier labs have closed most of the visible quality gap with US labs and advertise dramatically lower token prices. For anyone actually choosing a model, that raises two concrete questions:

1. **Are the Chinese flagships genuinely as good as the Western frontier on hard reasoning** — or do they win headline averages while quietly failing when problems get difficult?
2. **Is the cheap token price a real economic advantage** — or is it eaten by higher token consumption and slower wall-clock latency?

The honest way to answer both is a **cost-matched** comparison: pit the Chinese flagships against Western budget models at a comparable price point (the trade a buyer on a budget actually faces), with the two Western frontier models included only as a quality ceiling. And the honest way to read the answer is **by difficulty** — because, as this report shows, a single average conceals the only thing that matters.

Middle-school competition math is an ideal probe. The AMC 8 is the U.S. national contest for students in **grade 8 and below** (AJHSME is its junior-high predecessor); every problem has one unambiguous multiple-choice answer, so grading is objective; the problems span a wide, well-calibrated difficulty range; and they reward reasoning over recall — you cannot memorise your way through them.

---

2. How the test was run

2.1 The 16 models, in three cost blocs

Bloc	Models	Role
Chinese flagship (8)	DeepSeek-V4-Pro, Qwen3.7-Max, Kimi-K2.6, GLM-5.1, MiniMax-M2.7, ERNIE-4.5-VL, Seed-2.0-Lite, Step-3.7-Flash	the labs' top / near-top public models
Western budget (6)	Claude Haiku 4.5, GPT-5.4-mini, GPT-5.4-nano, Gemini 3.1 Flash-Lite, Grok 4.3, Llama-4-Maverick	cost-matched to the Chinese flagships
Western frontier (2)	Claude Opus 4.8, Claude Sonnet 4.6	accuracy-ceiling reference

The comparison is deliberately not like-for-like on capability tier; it is like-for-like on **price**. The

buyer's question is "for the money I'd spend on a Western budget model, can I get a Chinese flagship — and is it better?" Non-Anthropic models were served via OpenRouter, Anthropic models via the native API. (Routing logs show most Chinese models are served from **Western** GPU clouds — DeepInfra, Together, Novita, Parasail, Chutes — so the latency results below reflect each model's token economy and its provider's throughput, **not** Chinese domestic compute.)

2.2 Problems and difficulty tiers

Problems were sampled at run time from the **text-only AMC 8 / AJHSME pool: 700 problems (AMC 8: 451, AJHSME: 249)**, each with a verified answer key. (The underlying store also contains Math Kangaroo and AoPS material; those were **excluded** — only contest ∈ {AMC 8, AJHSME} enters the pool, and the 263 distinct problems actually served were all AMC 8 / AJHSME.) **Text-only:** any problem whose solution needs a figure was dropped so no model was penalised for vision.

Difficulty bands are assigned **by problem number** — the position within the contest, where #1 is easiest and #25 hardest:

- **easy** — earliest problems, ≈ #1–10
- **medium** — early-to-mid, ≈ #6–16
- **hard** — mid-to-late, ≈ #11–20
- **stretch** — the contest's hardest problems, ≈ #20–25 (this is AMC 8's own hard tail, **not** a harder competition)

2.3 The "all-at-once" harness, and grading

Each round, every model received **the same 12 problems in a single API call** and was told to show full numbered step-by-step working for each, ending in a line ANSWER <n>: X. This is deliberate: it tests batch reasoning under load, and a model that **truncates or exhausts its reasoning budget mid-batch loses every remaining answer** — which we count as a real robustness result, not an error to retry away. We graded **only the final letter** against the key (exact match). The output budget was raised to 64K tokens and the deadline to 15 minutes specifically so slow, verbose models would not be cut off — we wanted their true latency.

2.4 Scale, and how to read the numbers

28 rounds · 16 models · 4,896 model×problem attempts (4,740 graded after errors; 263 distinct problems). The 14 cost-matched models accumulated 264–336 questions each; the two frontier models 96 each (a focused 8-round arm, 2 rounds per tier). Total metered API spend ≈ **$6.80**; reasoning-quality judging was done on a flat-rate subscription.

Three metrics recur throughout, all per question:

- **accuracy %** — fraction of answers correct (the bar for a 5-choice problem is 20% by guessing).
- **latency** — wall-clock seconds, computed as the single batched call's time ÷ 12.
- **¢ / correct** — cents per correct answer = (cents per question) ÷ accuracy. This is the metric that matters to a buyer, because it charges you for both verbosity and wrong answers. We also report **completion %** (fraction of attempts that returned a gradable answer) and **tokens / question** (output verbosity).

2.5 Limitations (read before quoting)

- **Single domain.** Middle-school competition math. Says nothing about coding, long-context, writing, agentic, or multimodal work.
- **Answer-only grading.** A model can reach the right letter by a flawed path or a lucky guess; §7 inspects reasoning quality qualitatively, but the headline numbers are answer-accuracy.

- **Unequal N.** Frontier models have 96 attempts vs ≈300 for the cost-matched set, so their figures have wider error bars (consistent with the main arm where they overlap).
- **Provider variance.** OpenRouter routes to whichever provider is up; latency partly reflects provider load. Mitigated with retries, not pinned.
- **List prices**, at run time, excluding caching/batch discounts.

3. The pooled headline — and why it misleads

Here is the conventional leaderboard: every model, all difficulties pooled, sorted by accuracy.

Table 1 — per-model summary, all tiers pooled.

#	Model	Org	Tier	Acc %	n	Cmpl %	Latency	¢/Q	¢/correct	tok/Q
1	Qwen3.7-Max	CN	flagship	**99**	324	96	10.3 s	0.274	0.276	696
2	Seed-2.0-Lite	CN	lite	**99**	312	93	25.3 s	0.122	0.124	597
3	DeepSeek-V4-Pro	CN	flagship	**98**	336	100	15.1 s	0.075	**0.077**	818
4	GLM-5.1	CN	flagship	**98**	300	89	9.4 s	0.172	0.176	529
5	Claude Opus 4.8	US	frontier	**98**	96	100	**1.6 s**	0.410	0.419	**137**
6	Kimi-K2.6	CN	flagship	**98**	336	100	17.7 s	0.361	0.370	1037
7	Step-3.7-Flash	CN	flash	**98**	324	96	8.0 s	0.217	0.222	1869
8	Claude Sonnet 4.6	US	frontier	95	96	100	2.4 s	0.220	0.232	124
9	MiniMax-M2.7	CN	flagship	92	264	79	9.5 s	0.115	0.124	933
10	GPT-5.4-mini	US	budget	90	336	100	1.3 s	0.108	0.120	224
11	Claude Haiku 4.5	US	budget	90	336	100	1.6 s	0.120	0.133	218
12	GPT-5.4-nano	US	nano	89	336	100	2.4 s	0.042	0.048	322
13	ERNIE-4.5-VL	CN	flagship	88	336	100	15.3 s	0.094	0.108	717
14	Grok 4.3	US	flagship-mid	87	336	100	1.8 s	0.078	0.090	258
15	Gemini 3.1 Flash-Lite	US	budget	85	336	100	2.1 s	0.109	0.128	709
16	Llama-4-Maverick	US	open-mid	84	336	100	8.9 s	0.021	**0.025**	325

Read this table on its own and you would conclude the field is tightly bunched: fourteen of sixteen models land between 84% and 99%, separated by single points. **That conclusion is wrong** — or rather, it is an artifact of averaging easy problems (where everyone scores ≈95%) together with hard ones (where the field fans out from 65% to 100%). The pooled accuracy column is mostly measuring how many easy problems were in the mix.

Two things in the table do survive pooling and preview the rest of the report: **latency** (every Western model is 1–3 s; the accurate Chinese models are 8–25 s) and **token economy** (Opus reaches 98% on 137 tokens/question; Kimi needs 1,037; Step 1,869 — the same answers, 8–14× the words). Everything else is best understood tier by tier, which is what §§4–6 do.

> **Takeaway (read this first).** The pooled leaderboard is the wrong tool. Below ≈90% the differences are real; above it, pooled accuracy mostly reflects problem mix. The signal is in the difficulty breakdown — which is why the rest of this report is organised by tier.

4. Accuracy across the difficulty range

Accuracy is where difficulty matters most, because the whole field starts in the same place and ends up in two very different places.

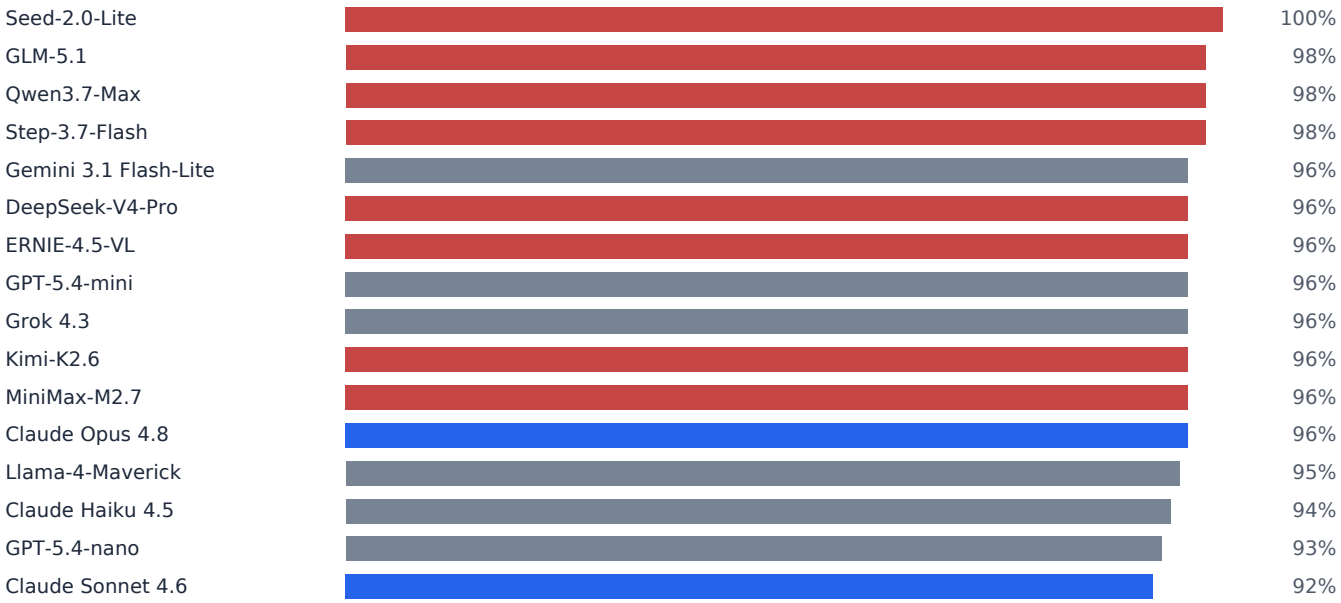
Table 2 — accuracy % by tier (the deterioration matrix).

Model	Org	easy	medium	hard	stretch
Qwen3.7-Max	CN	98	100	100	**100**
DeepSeek-V4-Pro	CN	96	98	99	**100**
Claude Opus 4.8	US	96	100	96	**100**
Kimi-K2.6	CN	96	98	96	**100**
Step-3.7-Flash	CN	98	96	97	99
GLM-5.1	CN	98	100	95	99
Seed-2.0-Lite	CN	100	100	99	95
MiniMax-M2.7	CN	96	97	77†	96
Claude Sonnet 4.6	US	92	96	100	92
GPT-5.4-mini	US	96	95	89	80
Claude Haiku 4.5	US	94	94	93	79
GPT-5.4-nano	US	93	90	90	81
ERNIE-4.5-VL	CN	96	90	95	**68**
Grok 4.3	US	96	95	83	71
Gemini 3.1 Flash-Lite	US	96	85	90	70
Llama-4-Maverick	US	95	89	87	**66**

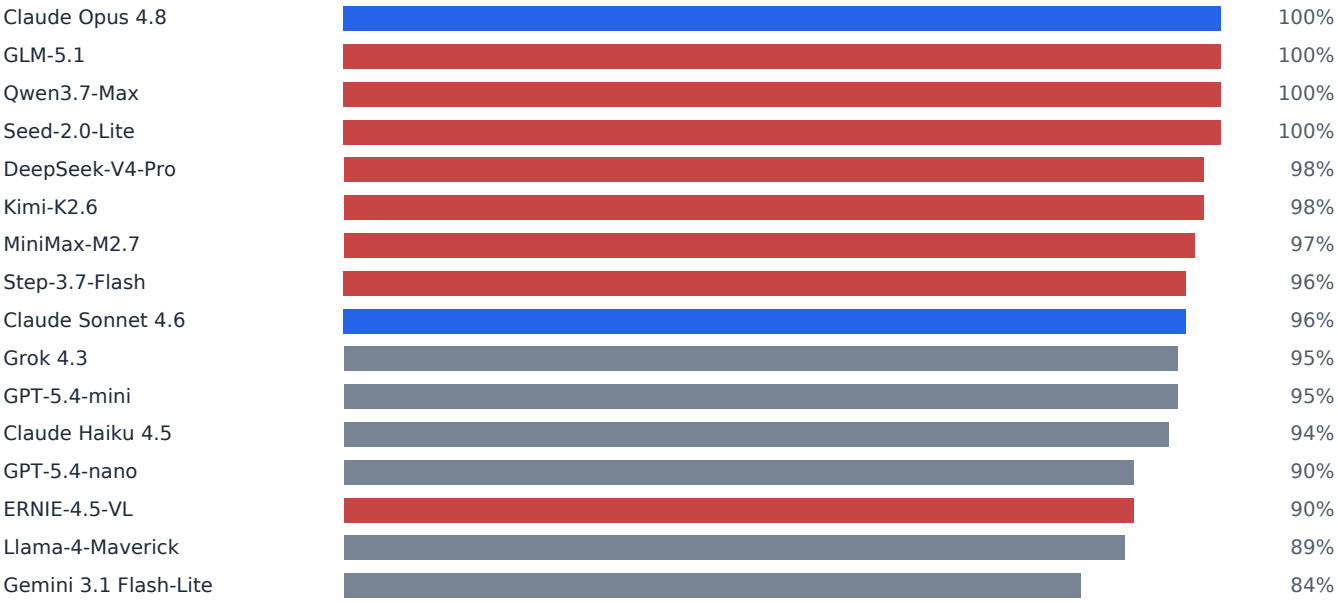
† MiniMax's hard-tier dip coincides with reasoning-budget exhaustion on some rounds (only 71% completion there).

Watch the field reorder itself one tier at a time — a flat wall on **easy** (fifteen of sixteen models within 7 points), holding firm through **medium** and **hard**, then a clean cliff on **stretch** (a 34-point spread):

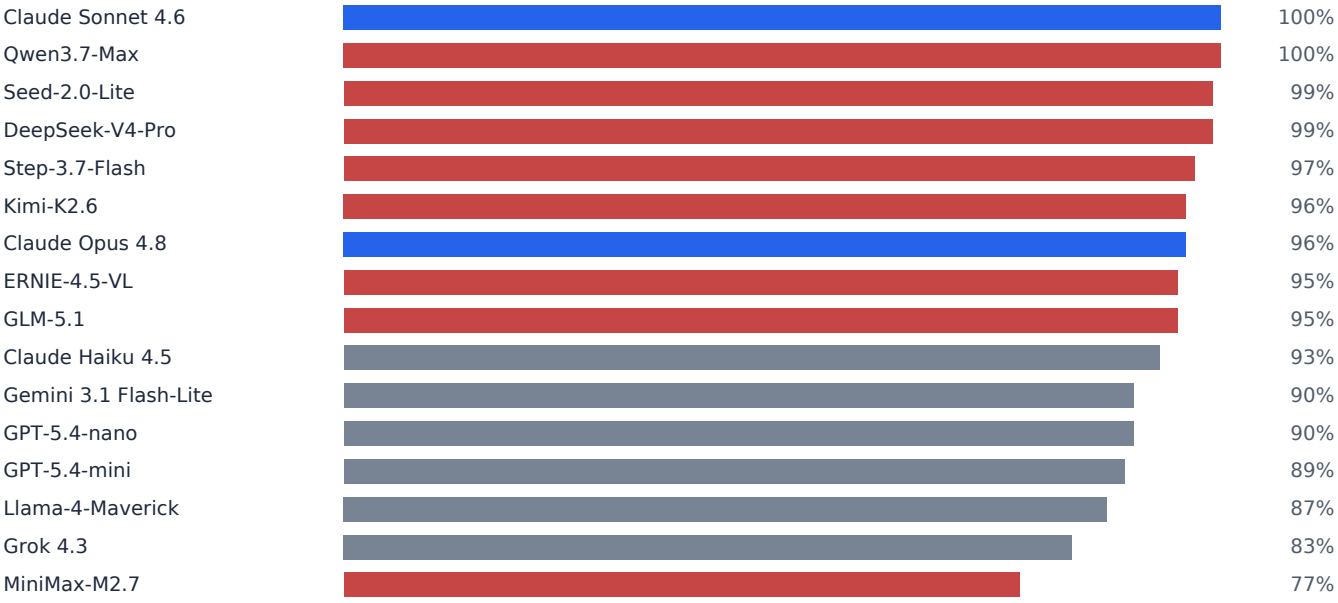
EASY tier — accuracy % (a flat wall: everyone is right)



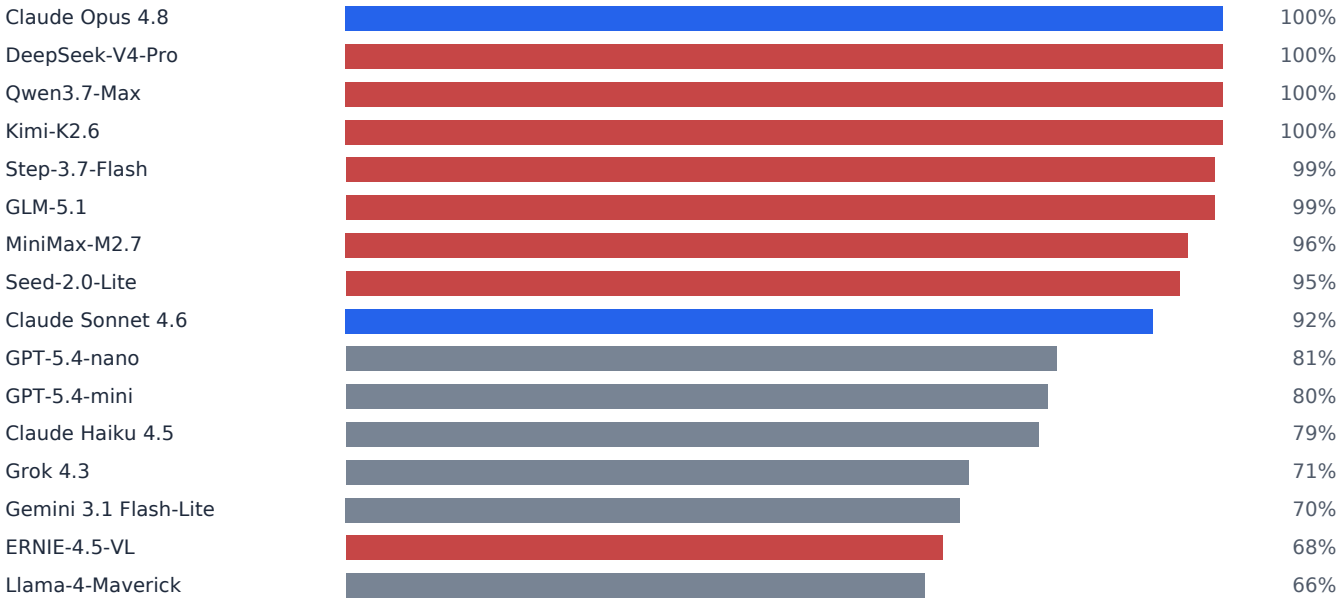
MEDIUM tier — accuracy %



HARD tier — accuracy %



STRETCH tier — accuracy % (a cliff: the field splits in two)



Aggregated, the divergence is unmistakable:

Bloc	easy	medium	hard	stretch	easy→stretch
Western frontier (Opus/Sonnet)	94	98	98	96	**+2** (none)
Chinese flagship	97	97	95	94	**-3** (negligible)
Western budget	95	91	89	**74**	**-21** (collapse)

The Chinese flagships and the Western frontier behave the same way — they hold the line, four of them actually scoring 100% on the hardest tier. The cost-matched Western budget bloc is the only group that falls apart, losing 15–30 points each from easy to stretch. Two Chinese names break ranks: **ERNIE** (96→68) and the budget-exhausting **MiniMax** behave like budget models on the hardest problems — proof that "being a real flagship," not "being Chinese," is the variable that predicts holding up.

> **Takeaway (accuracy).** On easy problems accuracy is uninformative — pick on something else. On hard problems, the order is frontier ≈ Chinese-flagship ≫ Western-budget. If your workload contains hard problems and you choose on a pooled average, you will overrate the cheap models badly.

5. Latency across the difficulty range

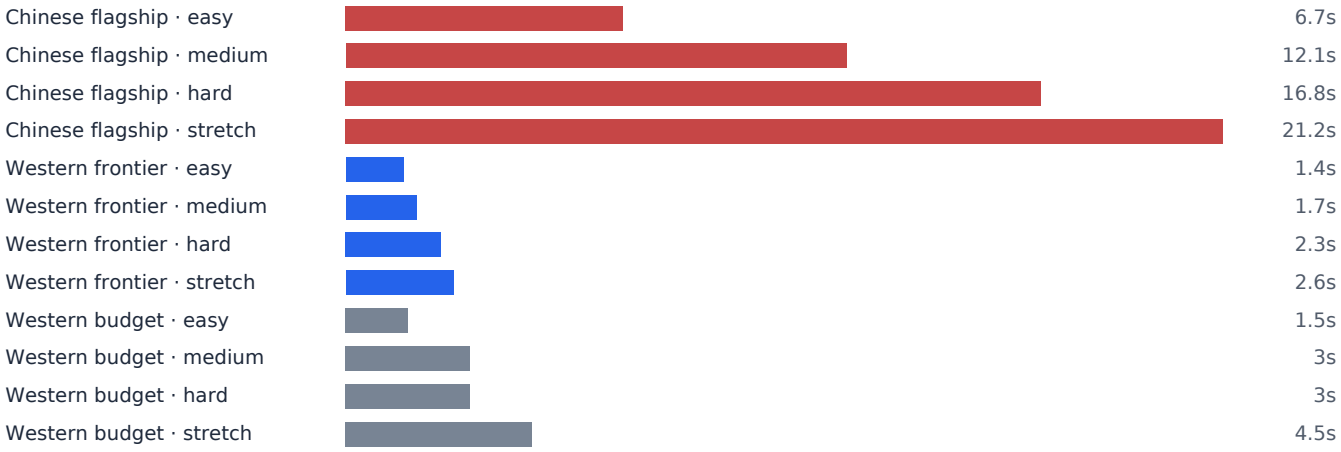
The pooled latency column already told us Western models answer in 1–3 s and accurate Chinese ones in 8–25 s. The tier breakdown adds the more useful fact: **latency is not a fixed property — it is elastic to difficulty, and the elasticity differs sharply by bloc.**

Table 3 — latency (seconds) by tier, per model (sorted slowest-overall first).

Model	Org	easy	medium	hard	stretch	ALL
Seed-2.0-Lite	CN	17.8	23.1	28.6	34.3	25.3
Kimi-K2.6	CN	9.1	17.2	17.6	27.0	17.7
ERNIE-4.5-VL	CN	3.9	9.0	20.0	28.2	15.3
DeepSeek-V4-Pro	CN	4.1	16.1	18.7	21.6	15.1
Qwen3.7-Max	CN	5.1	6.5	14.1	15.0	10.3
MiniMax-M2.7	CN	4.1	7.2	11.1	20.5	9.5
GLM-5.1	CN	6.1	8.2	11.3	13.1	9.4

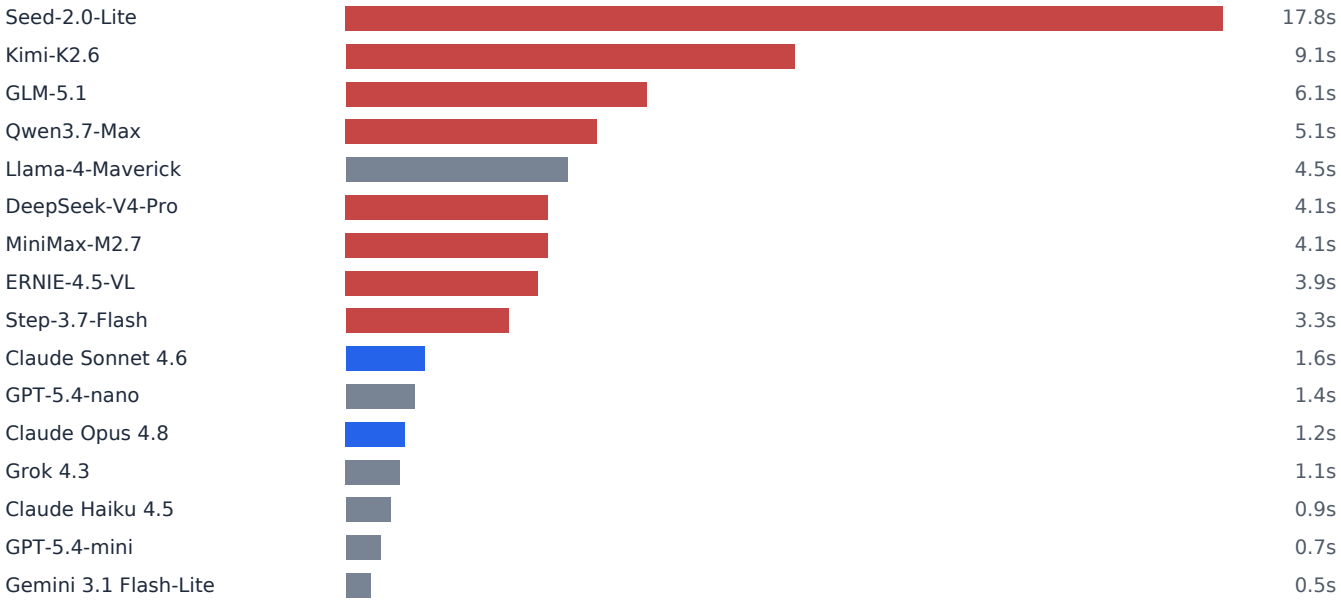
Model	Org	easy	medium	hard	stretch	ALL
Llama-4-Maverick	US	4.5	8.6	9.9	12.5	8.9
Step-3.7-Flash	CN	3.3	8.1	8.6	11.9	8.0
GPT-5.4-nano	US	1.4	2.0	2.4	3.8	2.4
Claude Sonnet 4.6	US	1.6	2.3	2.6	3.0	2.4
Gemini 3.1 Flash-Lite	US	0.5	2.8	0.8	4.4	2.1
Grok 4.3	US	1.1	2.0	2.1	2.1	1.8
Claude Opus 4.8	US	1.2	1.1	2.0	2.3	1.6
Claude Haiku 4.5	US	0.9	1.4	1.6	2.4	1.6
GPT-5.4-mini	US	0.7	1.1	1.3	2.0	1.3

Average latency by tier, per bloc — the Chinese "crawl" steepens with difficulty

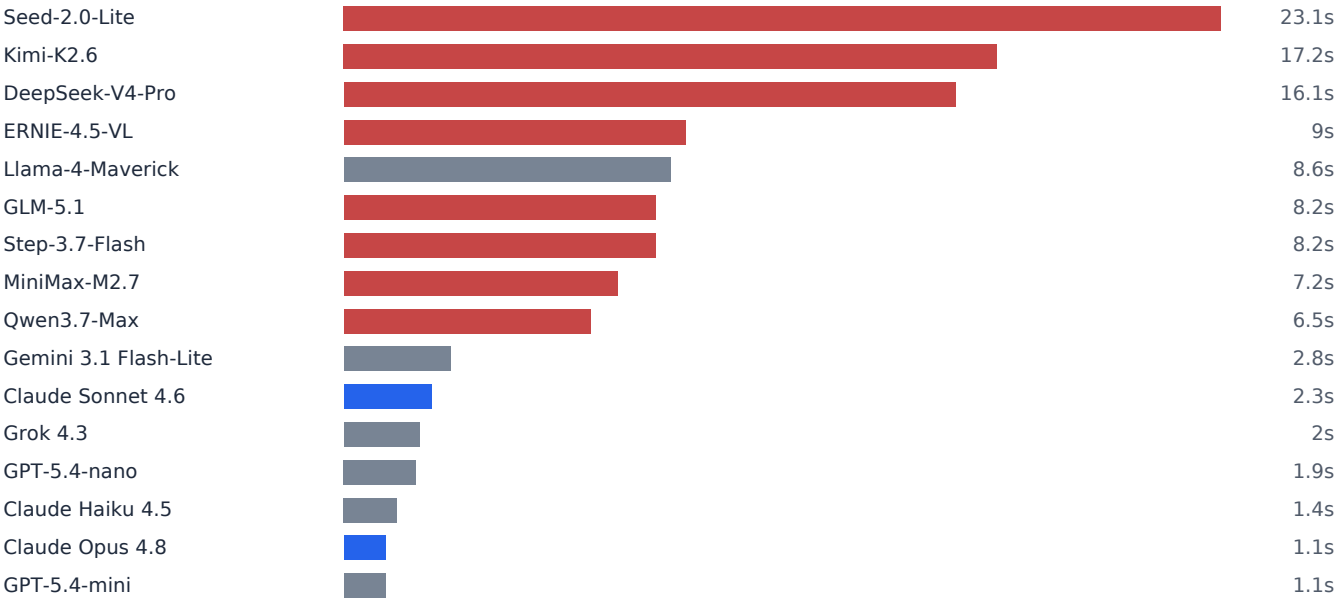


The frontier is **difficulty-inelastic**: Opus runs 1.2 s on easy and 2.3 s on stretch — essentially flat. Everyone else's clock climbs with difficulty, and the Chinese bloc climbs steepest, roughly **tripling** from ≈7 s to ≈21 s (individual models worse: Seed 18→34 s, ERNIE 4→28 s, DeepSeek 4→22 s). Tier by tier:

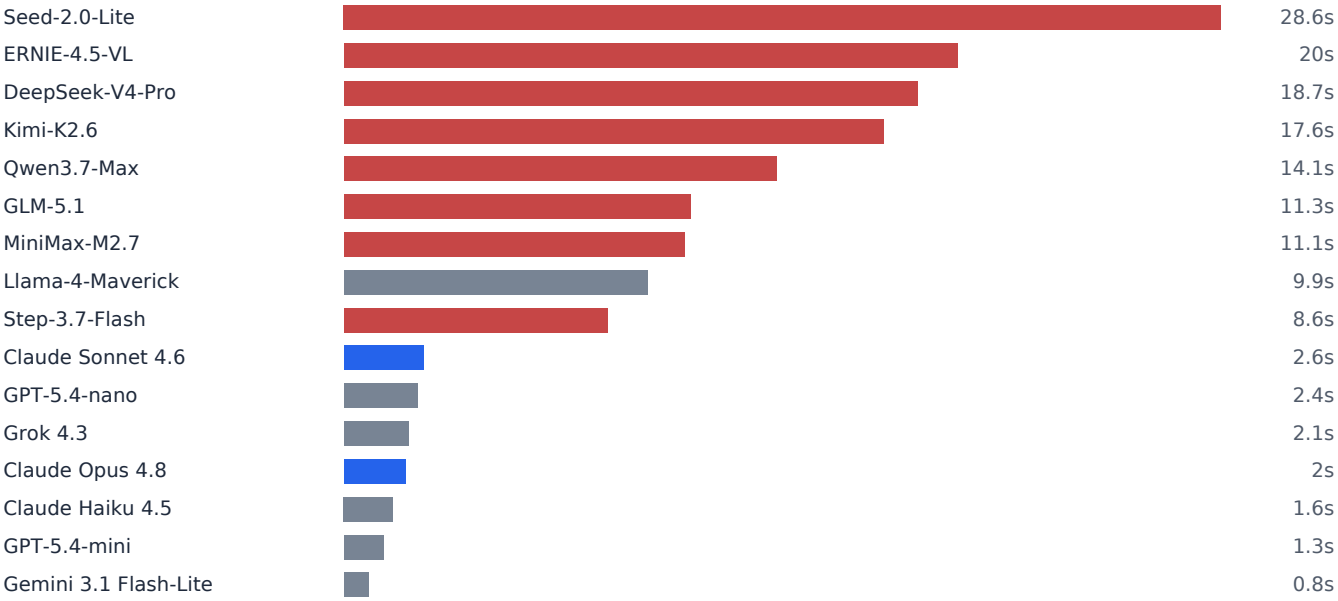
EASY tier — latency (s): even here, Chinese models are the slow ones



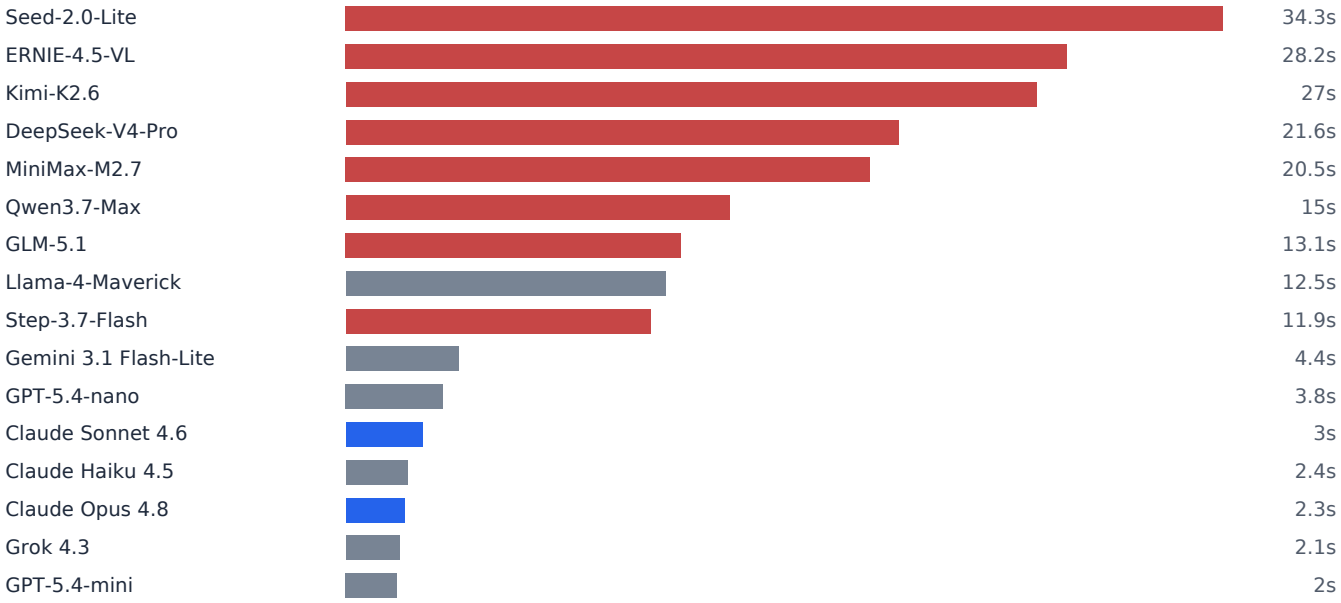
MEDIUM tier — latency (s)



HARD tier — latency (s)



STRETCH tier — latency (s): the Chinese flagships hit 12–34 s, the frontier stays ≈3 s



This sharpens the usual "Chinese models are slow" line into something more useful: they are slow **specifically on hard problems**. On easy work most of them answer in 3–6 seconds — annoying but tolerable. It is on the hard tier — exactly where a person would also slow down, and exactly where an interactive tool can least afford a stall — that they balloon to 12–34 seconds, a **7–13× wait** versus the frontier's ≈3 s. The mechanism is in §7: they get hard problems right by writing more, and writing more takes time.

> **Takeaway (latency).** If latency matters at all — chat, agents, anything interactive — the Western frontier is the only bloc that stays responsive on hard problems. The Chinese flagships are a batch/offline technology: fine when you can wait, disqualifying when you can't.

6. Cost across the difficulty range

This is the richest tier story, because cost-per-correct has **two moving parts**, and difficulty pushes on both:

> **¢ per correct = (¢ per question) ÷ accuracy.**

There are therefore two ways for a model to become expensive per correct answer: **inflate the numerator** (burn tokens) or **shrink the denominator** (get answers wrong). On easy problems both terms are benign for everyone. As difficulty rises they diverge — and watching which term moves for which bloc is the whole story.

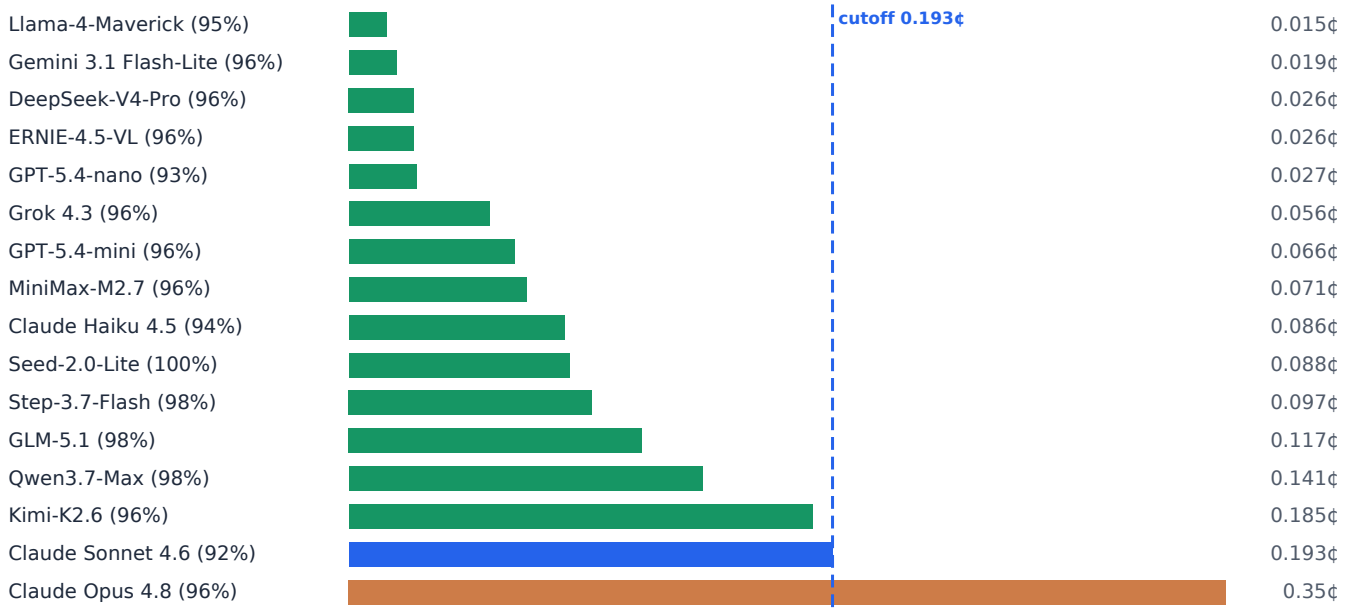
Table 4 — cents per correct answer by tier, per model (sorted cheapest-overall first).

Model	Org	easy	medium	hard	stretch	ALL
Llama-4-Maverick	US	0.015	0.020	0.022	0.050	0.025
GPT-5.4-nano	US	0.027	0.036	0.048	0.084	0.048
DeepSeek-V4-Pro	CN	0.026	0.085	0.099	0.095	**0.077**
Grok 4.3	US	0.056	0.082	0.102	0.132	0.090
ERNIE-4.5-VL	CN	0.026	0.064	0.130	0.250	0.108
GPT-5.4-mini	US	0.066	0.095	0.122	0.213	0.120
Seed-2.0-Lite	CN	0.088	0.110	0.137	0.177	0.124
MiniMax-M2.7	CN	0.071	0.090	0.203	0.192	0.124
Gemini 3.1 Flash-Lite	US	0.019	0.160	0.032	0.360	0.128
Claude Haiku 4.5	US	0.086	0.111	0.134	0.217	0.133

Model	Org	easy	medium	hard	stretch	ALL
GLM-5.1	CN	0.117	0.148	0.184	0.271	0.176
Step-3.7-Flash	CN	0.097	0.229	0.236	0.329	0.222
Claude Sonnet 4.6	US	0.193	0.222	0.246	0.267	0.232
Qwen3.7-Max	CN	0.141	0.175	0.379	0.391	0.276
Kimi-K2.6	CN	0.185	0.342	0.523	0.428	0.370
Claude Opus 4.8	US	0.350	0.310	0.503	0.512	0.419

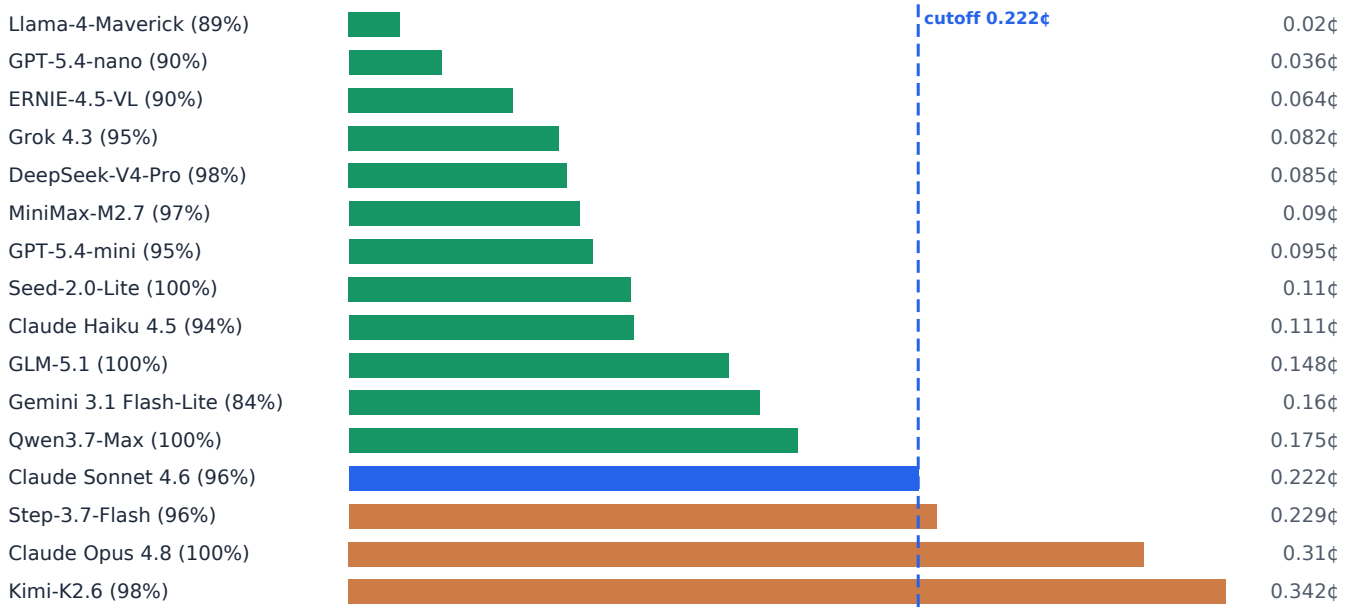
On **easy** problems the cheap models win outright and the frontier is simply wasteful — note Opus at 0.35 ¢ for the same 96% that DeepSeek delivers at 0.026 ¢ and Gemini at 0.019 ¢. The Sonnet line (0.193 ¢) sits near the expensive end:

EASY tier — ¢ per correct (cheap models win; the frontier is wasted money)

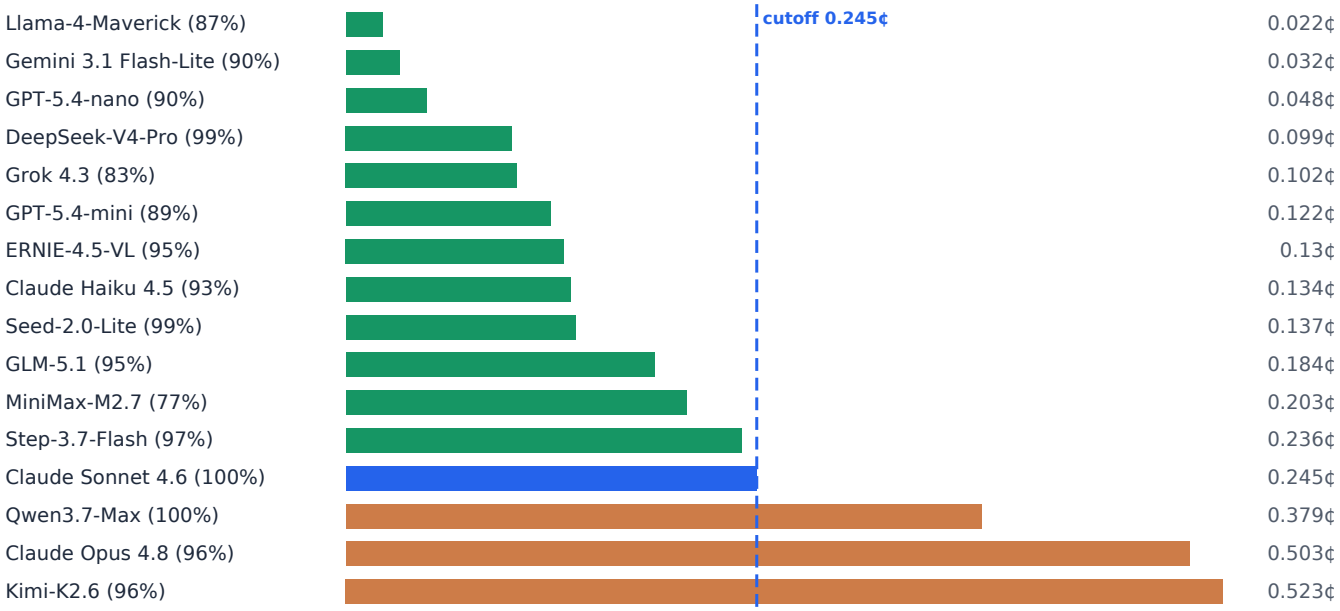


Through **medium** and **hard**, DeepSeek pulls clear of the pack as the budget models' accuracy starts to slip and the verbose flagships' token bills mount:

MEDIUM tier — ¢ per correct (Sonnet = cutoff)

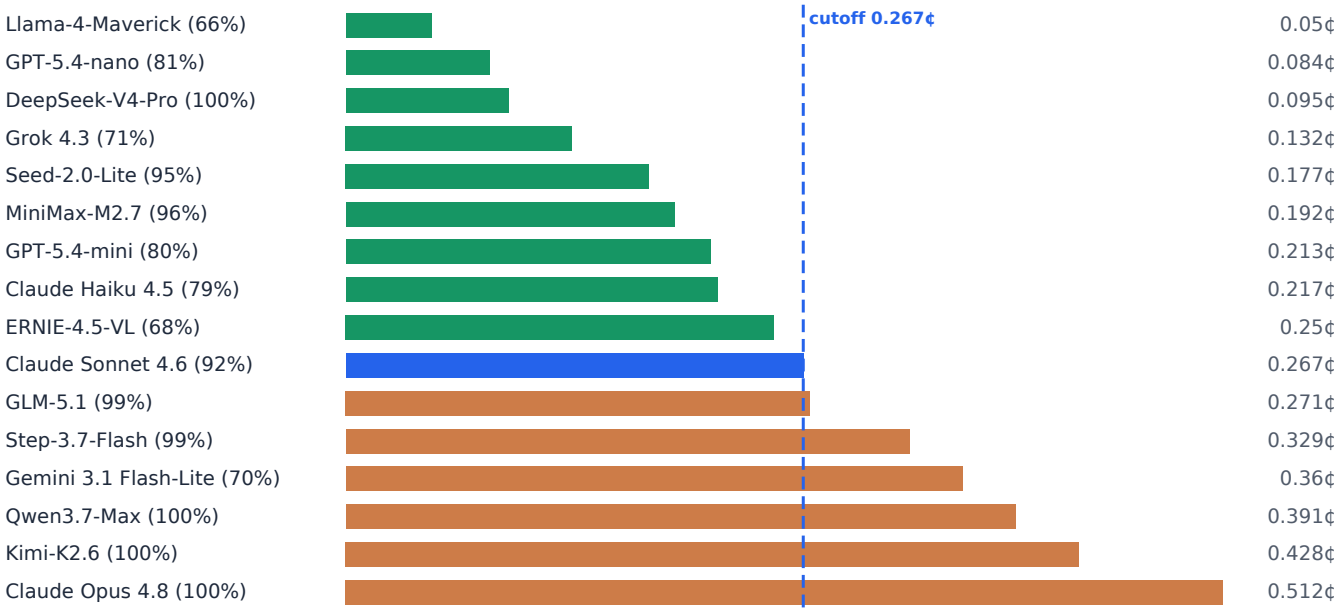


HARD tier — ¢ per correct (Sonnet = cutoff)



By **stretch** the picture has turned over, and the reasons split by bloc:

STRETCH tier — ¢ per correct (two ways to get expensive; DeepSeek avoids both)



Trace the two doors across Table 4:

- **Numerator blow-ups (pure token burn).** Kimi's ¢/correct climbs 0.185 → 0.523 (easy→hard) and Opus's 0.350 → 0.512 (→stretch) — not because they get answers wrong (both stay ≈100% accurate) but purely because they **write more** on harder problems. Accurate, and increasingly expensive.
- **Both doors at once (the budget tier).** On hard and stretch the budget models spend more tokens and miss more answers, so cost-per-correct is hit from both sides. Gemini is the worst case: on stretch it balloons output (107 → 1,667 tokens) and falls to 70% accuracy, so its ¢/correct leaps ≈19× (0.019 → 0.36). Even where tokens are restrained, the denominator alone bites — when only ≈70% of attempts are right, each correct answer costs ≈1.4 attempts before any token inflation.
- **The one model that does neither badly. DeepSeek-V4-Pro** keeps verbosity moderate and accuracy ≈100%, so it stays the **cheapest accurate model at every tier** — 0.026 ¢ on easy, rising only to ≈0.095 ¢ on hard and stretch, far below any model of comparable

accuracy. It is cheap on easy problems and still the value pick on hard ones.

The bloc-level consequence is that the much-advertised Chinese price advantage is itself a **decreasing function of difficulty**. The Chinese flagships burn 4.1× the frontier's tokens on easy problems but 8.4× on stretch, so their cost as a share of the frontier's rises steadily:

Tier	Chinese tokens vs frontier	Chinese ¢/Q as share of frontier
easy	4.1×	**0.36×** (64% cheaper)
medium	7.4×	0.58×
hard	7.3×	0.63×
stretch	8.4×	**0.69×** (only 31% cheaper)

Chinese flagship cost as a share of frontier cost — the advantage shrinks with difficulty

And measured per correct answer rather than per token, the advantage doesn't just shrink — for the verbose flagships it **disappears**: by the hard tier, Qwen (0.379 ¢) and Kimi (0.523 ¢) cost more per correct answer than Sonnet (0.246 ¢), and Kimi rivals Opus. "Cheap Chinese tokens" is true per token and, for the chatty models on hard problems, false per answer.

> **Takeaway (cost).** Cost-per-correct is driven by both verbosity and accuracy, and difficulty attacks through both: the verbose flagships are sunk by token burn (accuracy intact), the budget tier by wrong answers and extra tokens. Only **DeepSeek** keeps both in check and stays cheap-and-accurate at every tier — the single genuine value outlier in the study. **Kimi** is the opposite — frontier-level cost with none of the frontier's speed.

7. One mechanism behind all three: brute force vs. finesse

Accuracy parity, the latency crawl, and the cost erosion are three views of a single underlying fact: **the two blocs reach the same answers by opposite routes.**

The Western frontier is **terse**. Output tokens per question barely rise with difficulty — Opus/Sonnet go from ≈103 on easy to ≈160 on stretch (+55%) — and accuracy holds anyway. This is genuine reasoning efficiency: Opus solves a stretch problem correctly in fewer tokens than a Chinese flagship spends on an easy one.

The Chinese flagships are **verbose by design**. Their output scales from ≈420 tokens on easy to ≈1,340 on stretch (+218%); Step reaches 2,807 on stretch. They get hard problems right by thinking out loud at length — and it works (their accuracy barely drops). But every one of those extra tokens costs a fraction of a second and a fraction of a cent, which is exactly why they are slow (§5) and why their per-answer cost climbs (§6). It is brute force, and it is effective; it is simply not free.

The Western budget models try to scale tokens too (155 → 618, +298%) but **without the competence to convert effort into correctness** — their accuracy falls anyway. Effort without finesse.

Importantly, the slowness is a design choice, not a hardware deficit: most of these Chinese models run on the same Western GPU clouds as everyone else. They spend test-time compute generously because that is how they were trained to be accurate.

One problem, three styles. A stretch problem — "There are 24 four-digit numbers using each of

2, 4, 5, 7 once; exactly one is a multiple of another. Which?" (answer: D) — shows the routes concretely:

- **Opus 4.8 (correct, terse):** states the divisibility constraint, narrows the candidates, lands on D in ≈3 sentences (≈140 tokens). No wasted motion — finesse.
- **DeepSeek-V4-Pro (correct, exhaustive):** works it systematically, enumerating constraints at length before converging on D — rigorous, correct, several times longer. Brute force.
- **Grok 4.3 (correct):** reaches D cleanly — a budget model can get hard problems, just less reliably across the tier.
- **Llama-4-Maverick (wrong → C):** sets up correctly, slips mid-derivation, commits confidently to C. The reasoning looks fine — the danger of answer-only confidence in budget models on hard problems.

> **Takeaway (mechanism).** Same destination, opposite vehicles: the frontier pays in dollars-per-token for a terse, fast solution; the Chinese flagships pay in seconds-and-tokens for a verbose one. The budget models pay too — and still get lost on the hard roads.

---

8. Choosing a model: the Sonnet 4.6 cutoff, by difficulty

The most useful reference point in the study is **Claude Sonnet 4.6** — fast (2.4 s), cheap per correct answer (0.232 ¢), accurate (95%), and notably faster than every Chinese model and cheaper-per-correct than most of them. Treat it as the cutoff: **a model only earns a place over Sonnet if it beats Sonnet on an axis you actually care about.** Three gates:

1. **Cheaper per correct and at least as accurate (≥95%)?** Only three pass: **DeepSeek** (0.077 ¢, 98%), **Seed** (0.124 ¢, 99%), **GLM** (0.176 ¢, 98%) — the defensible cost-driven upgrades, all of which cost you 9–25 s of latency.
2. **Faster than Sonnet at comparable accuracy?** Only the frontier and a couple of budget models; the one that is both faster and more accurate is **Opus** (1.6 s, 98%). Nothing Chinese is remotely in this gate.
3. **Fails both?** Dominated by Sonnet. This is where **Qwen, Kimi, Step, ERNIE, MiniMax** land — slower than Sonnet and, for the verbose ones, no cheaper per correct answer.

But the cutoff is not static — **the right pick depends on difficulty**, which is the through-line of this whole report:

Your workload	Best pick	Why
Easy / bulk / offline	cheapest competent model — **Gemini Flash-Lite, GPT-nano, or DeepSeek**	accuracy is a non-issue at ≈95%; pay the least. A flagship is wasted money.
Mixed, cost-sensitive, latency-tolerant	**DeepSeek-V4-Pro**	flagship accuracy at the lowest effective cost (0.077 ¢), nearly flat across tiers; tolerate ≈15 s
Hard / interactive / latency matters	**Opus 4.8** (or Sonnet)	only bloc that stays accurate and < 3 s on hard problems; nothing Chinese is under 8 s
Balanced default, no caveats	**Sonnet 4.6**	the cutoff itself — fast, cheap, 95%
Hard problems on a Chinese flagship	**reconsider**	12–34 s latency makes them impractical for anything interactive, regardless of accuracy

> **Takeaway (decision).** Benchmark on the hard tail, measure cost-per-correct and latency per tier, and use Sonnet as the bar. Two models clear it: DeepSeek on cost, Opus on speed. On easy work, ignore all of this and buy the cheapest thing that's competent.

---

9. The Chinese field, internally

Eight Chinese models, ranked by accuracy. The race at the top is a near-tie; the real differentiation is **speed, token discipline, and robustness** — and the tier breakdowns above are what separate them.

Rank	Model	Lab	Acc %	Latency	¢/correct	tok/Q	Verdict
1.	**DeepSeek-V4-Pro**	DeepSeek	98	15.1 s	**0.077**	818	**Overall champion.** Frontier accuracy that holds (100% on stretch), 100% completion, cheapest effective cost at every tier. The one model that clears the Sonnet cutoff comfortably.
2.	**Qwen3.7-Max**	Alibaba	**99**	10.3 s	0.276	696	**Accuracy king** and the most token-disciplined of the high-accuracy Chinese set — but its ¢/correct crosses above Sonnet by the hard tier.
3.	**GLM-5.1**	Zhipu	98	**9.4 s**	0.176	529	**Speed/efficiency pick** — fastest of the accurate Chinese models and the leanest (529 tok). Slight wobble on hard (95%); some completion loss.
4	Step-3.7-Flash	StepFun	98	8.0 s	0.222	1869	Fast and accurate but **wildly verbose** (1,869 tok/Q, 2,807 on stretch) — pays for its speed in token volume.
5	Kimi-K2.6	Moonshot	98	17.7 s	0.370	1037	**The standout disappointment.** Accurate and reliable, but the priciest Chinese model per correct answer (near Opus), second-slowest, and among the most verbose — wins on no axis.
6	Seed-2.0-Lite	ByteDance	**99**	**25.3 s**	0.124	597	**The sleeper** — 99% from a model marketed "lite," and cheap — at the cost of being the **slowest model in the study** (34 s on stretch).
7	MiniMax-M2.7	MiniMax	92	9.5 s	0.124	933	**Robustness problem** — only 79% completion; exhausts its reasoning budget mid-batch on hard rounds (77% there).
8	ERNIE-4.5-VL	Baidu	88	15.3 s	0.108	717	**Flagship in name only** on hard problems — collapses to 68% on stretch while still taking 28 s. The worst of both worlds.

- **DeepSeek, Qwen and GLM are the genuine value frontier** — each defensible as "best Chinese model" on a different axis (cost / accuracy / efficiency).
- **Kimi is the cautionary tale**: matching the cluster on accuracy is not enough when a faster,

cheaper option (Sonnet, or DeepSeek) matches it.

- **MiniMax and ERNIE are flagships in name but not in hard-problem behaviour** — one fails on robustness, the other on capability.
- **ByteDance's "Lite" model is the sleeper** — top-tier accuracy, bottom-tier speed. The naming misleads in both directions.

10. Commercial coda — where the revenue actually is

A benchmark measures capability, but capability and revenue are different axes, and it is worth saying plainly which column maps to money. **The cost column measures usage; the latency column gates revenue.**

Revenue in this market follows a Pareto rule. A **minority of price-insensitive buyers — the ones who want the latest and greatest — throw off most of the revenue**, and they buy on speed × quality, not token price. A **price-sensitive majority keeps shopping for the cheapest tokens**: enormous usage, thin margin, no loyalty, a permanent willingness to switch the moment someone is a hair cheaper. **Usage is not revenue.** The §6 cost analysis, where DeepSeek and the budget models shine, is largely a map of the usage axis — the high-volume, low-margin lane. The revenue-dense lane is the one §5 measures: low latency, and high accuracy on the problems that are genuinely hard.

In this benchmark only the Western frontier occupies that cell. Opus and Sonnet are difficulty-**inelastic** — ≈1.2–3.0 s at every tier while holding 95–100% on hard problems — whereas every accurate Chinese model is difficulty-elastic, climbing to 12–34 s exactly on the hard inputs. And **latency is a product gate, not a dial**: a customer-facing product, or a human-in-the-loop agent, has to size its timeout for the worst input it will see, so a model that blows to 27–34 s on a single hard step cannot sit behind that UX at any price. Agentic loops make it worse — per-step latency compounds, so a 20-step run finishes in ≈40 s at frontier speed versus **5–7 minutes** on a flagship spending 15–21 s a step. Those compounding, interactive workloads are exactly the ones scaling fastest and paying most.

This is why **the real competitive fight is rate limits, not sticker price.** For production traffic the binding constraint is correct-answers-per-second at peak — governed by throughput and committed capacity (RPM/TPM), not cents per token. Two forces compound. Effective throughput ≈ concurrency ÷ latency, so a 21 s request ties up a serving slot ≈10× longer than a 2 s one — a slow model's cost-to-serve at an interactive SLA is high even when its list price is low. And because the Chinese flagships buy their accuracy with 5–10× more output tokens, and TPM ceilings are denominated in tokens, the same workload burns the rate-limit budget 5–10× faster. **The cheap-token win and the capacity loss are the same coin.**

The honest caveat — and it is real — is that "latency-tolerant = hobbyist" is too strong. A genuinely enterprise-funded, latency-tolerant pool exists: overnight analytics, RAG / eval / data-labeling and synthetic-data pipelines, long-horizon async agents. (Anthropic's own 50%-off Batch tier is revealed-preference proof that serious time-insensitive spend is worth pricing for.) But that pool is **lower-margin** — price-elastic, shopped to the floor, partly fenced off by data-residency rules that bar Chinese-hosted models, and competing against the frontier's own discounted batch endpoints. It monetizes, thinly.

The demand side maps onto two axes — willingness-to-pay and latency-need — and the revenue pools in one corner:

Willingness-to-pay ↓ / Latency need →	Latency-tolerant (batch OK)	Latency-sensitive (interactive)
High (won't switch on price)	Enterprise batch — analytics, eval / RAG, async agents. Real spend, but price-shopped and lower-margin; served by frontier-batch endpoints or DeepSeek.	**The revenue pool (the money)** — customer-facing apps, synchronous agents. Pays a premium for speed and throughput. → **frontier**.
Low (chases cheapest tokens)	Hobbyists, indie & SME batch. High usage, thin margin, no loyalty. → **DeepSeek's home**.	Indie / SME interactive — wants fast and cheap, settles for budget models that crack on hard.

Mapped onto the models, the verdict is sharper than "Opus for speed, DeepSeek for cost":

- **The frontier (Opus, Sonnet) owns revenue** — the dense, price-insensitive, low-latency lane and the throughput premium enterprises pay to escape contention.
- **The verbose flagships (Kimi, Qwen) are squeezed from both sides** — too slow for the revenue lane, and, by the hard tier, more expensive per correct answer than Sonnet (0.370 ¢ and 0.391 ¢ vs 0.246 ¢), so they lose the commodity lane to DeepSeek too. No path.
- **DeepSeek-V4-Pro is the genuine winner-in-waiting.** It has already solved the hard half — frontier-grade accuracy at a commodity price (0.077 ¢/correct, ≈3× under Sonnet, ≈5× under Opus). Its one missing piece is latency. So it is a winner **conditional on three things**: (1) that it is sustainably profitable at these price points, not buying share; (2) that it can **serve the volume** without hitting the capacity/rate-limit wall that defines the real fight; and (3) that it **closes the latency gap** from ≈15–34 s toward interactive range. Clear all three and DeepSeek does not merely win the thin batch lane — it starts pressuring the revenue lane itself, because the frontier's moat here is speed, not accuracy, and speed is an engineering problem, not a capability one.

> **Takeaway (commercial).** Revenue is Pareto-distributed and sits at low-latency × high-accuracy — a cell only the frontier currently fills; cheap tokens win usage, not margin. The one model with a credible path from the usage lane into the revenue lane is **DeepSeek** — if it can stay profitable, scale capacity, and get fast. Everyone whose only edge is price is fighting over the lane with the least money in it.

11. Conclusions

1. **Difficulty is the benchmark.** Pooled accuracy is nearly useless here — fourteen of sixteen models sit at 84–99%. The field only resolves on the hard tail, and so do the cost and latency stories. Any evaluation that doesn't stress hard problems and report cost/latency per tier will conclude, wrongly, that the cheap models are good enough.
2. **Accuracy parity is real — and so is its limit.** Cost-matched, the best Chinese flagships (DeepSeek, Qwen, GLM, Kimi) match the Western frontier and beat the Western budget tier outright; on the hardest problems they hold while the budget tier collapses. The "Chinese models fail on hard problems" narrative is false for the true flagships — and true for ERNIE and MiniMax, which are flagships in name only.
3. **The frontier's edge is efficiency, not accuracy.** Opus reaches the same answers in ≈1/8 the tokens and stays under 3 s at every difficulty. If latency matters, nothing Chinese competes — they are a batch technology that becomes unusably slow (12–34 s) exactly on hard problems.
4. **"Cheap Chinese tokens" is a difficulty-dependent half-truth.** Genuine on easy work (≈⅓ of frontier cost); largely gone on hard work, where the verbose flagships' cost-per-correct reaches or exceeds Sonnet's. Cost-efficiency dies two ways — token burn (Kimi/Opus) or wrong answers (the budget tier) — and **DeepSeek-V4-Pro is the only model that avoids**

both at every tier, making it the study's clear value champion.

5. **Use Sonnet as the cutoff, then choose by difficulty.** Easy work: the cheapest competent model. Hard or interactive work: Opus for speed, DeepSeek for cost. Everything in between is dominated.
6. **Usage is not revenue (§10).** Revenue follows a Pareto rule and concentrates at low-latency × high-accuracy — the cell only the frontier fills — so commercially the speed result outweighs the cost result; competing on cheap tokens wins the high-volume, low-margin usage lane, where the real constraint is rate-limit capacity, not price. The one model with a credible path from that lane into the revenue lane is **DeepSeek** — if it can stay profitable at these prices, serve the volume, and close the latency gap.

Appendix A — full per-cohort tables

Every Table-1 parameter, split by tier. Within each cohort, models are sorted by accuracy then latency.

EASY cohort

Model	Org	Tier	Acc %	n	Cmpl %	Latency	¢/Q	¢/correct	tok/Q
Seed-2.0-Lite	CN	lite	100	84	100	17.8 s	0.088	0.088	426
Step-3.7-Flash	CN	flash	98	84	100	3.3 s	0.094	0.097	805
Qwen3.7-Max	CN	flagship	98	84	100	5.1 s	0.138	0.141	333
GLM-5.1	CN	flagship	98	84	100	6.1 s	0.114	0.117	341
Gemini 3.1 Flash-Lite	US	budget	96	84	100	0.5 s	0.019	0.019	107
GPT-5.4-mini	US	budget	96	84	100	0.7 s	0.063	0.066	125
Grok 4.3	US	flagship-mid	96	84	100	1.1 s	0.054	0.056	164
ERNIE-4.5-VL	CN	flagship	96	84	100	3.9 s	0.025	0.026	165
DeepSeek-V4-Pro	CN	flagship	96	84	100	4.1 s	0.025	0.026	239
MiniMax-M2.7	CN	flagship	96	84	100	4.1 s	0.069	0.071	555
Kimi-K2.6	CN	flagship	96	84	100	9.1 s	0.178	0.185	503
Claude Opus 4.8	US	frontier	96	24	100	1.2 s	0.336	0.350	108
Llama-4-Maverick	US	open-mid	95	84	100	4.5 s	0.014	0.015	211
Claude Haiku 4.5	US	budget	94	84	100	0.9 s	0.080	0.086	140
GPT-5.4-nano	US	nano	93	84	100	1.4 s	0.025	0.027	183
Claude Sonnet 4.6	US	frontier	92	24	100	1.6 s	0.177	0.193	97

MEDIUM cohort

Model	Org	Tier	Acc %	n	Cmpl %	Latency	¢/Q	¢/correct	tok/Q
Claude Opus 4.8	US	frontier	100	24	100	1.1 s	0.310	0.310	97
Qwen3.7-Max	CN	flagship	100	72	86	6.5 s	0.175	0.175	431
GLM-5.1	CN	flagship	100	84	100	8.2 s	0.148	0.148	449
Seed-2.0-Lite	CN	lite	100	84	100	23.1 s	0.110	0.110	538
DeepSeek-V4-Pro	CN	flagship	98	84	100	16.1 s	0.083	0.085	911
Kimi-K2.6	CN	flagship	98	84	100	17.2 s	0.334	0.342	958
MiniMax-M2.7	CN	flagship	97	72	86	7.2 s	0.087	0.090	706
Step-3.7-Flash	CN	flash	96	84	100	8.2 s	0.220	0.229	1900
Claude Sonnet 4.6	US	frontier	96	24	100	2.3 s	0.213	0.222	119
GPT-5.4-mini	US	budget	95	84	100	1.1 s	0.091	0.095	186
Grok 4.3	US	flagship-mid	95	84	100	2.0 s	0.078	0.082	261
Claude Haiku 4.5	US	budget	94	84	100	1.4 s	0.105	0.111	188
GPT-5.4-nano	US	nano	90	84	100	2.0 s	0.033	0.036	248
ERNIE-4.5-VL	CN	flagship	90	84	100	9.0 s	0.058	0.064	428

Model	Org	Tier	Acc %	n	Cmpl %	Latency	¢/Q	¢/correct	tok/Q
Llama-4-Maverick	US	open-mid	89	84	100	8.6 s	0.018	0.020	280
Gemini 3.1 Flash-Lite	US	budget	85	84	100	2.8 s	0.135	0.160	885

HARD cohort

Model	Org	Tier	Acc %	n	Cmpl %	Latency	¢/Q	¢/correct	tok/Q
Claude Sonnet 4.6	US	frontier	100	24	100	2.6 s	0.246	0.246	140
Qwen3.7-Max	CN	flagship	100	84	100	14.1 s	0.379	0.379	976
DeepSeek-V4-Pro	CN	flagship	99	84	100	18.7 s	0.098	0.099	1081
Seed-2.0-Lite	CN	lite	99	84	100	28.6 s	0.136	0.137	664
Step-3.7-Flash	CN	flash	97	72	86	8.6 s	0.230	0.236	1982
Kimi-K2.6	CN	flagship	96	84	100	17.6 s	0.504	0.523	1455
Claude Opus 4.8	US	frontier	96	24	100	2.0 s	0.483	0.503	164
ERNIE-4.5-VL	CN	flagship	95	84	100	20.0 s	0.124	0.130	956
GLM-5.1	CN	flagship	95	60	71	11.3 s	0.175	0.184	537
Claude Haiku 4.5	US	budget	93	84	100	1.6 s	0.124	0.134	227
Gemini 3.1 Flash-Lite	US	budget	90	84	100	0.8 s	0.029	0.032	177
GPT-5.4-nano	US	nano	90	84	100	2.4 s	0.043	0.048	330
GPT-5.4-mini	US	budget	89	84	100	1.3 s	0.109	0.122	225
Llama-4-Maverick	US	open-mid	87	84	100	9.9 s	0.019	0.022	295
Grok 4.3	US	flagship-mid	83	84	100	2.1 s	0.085	0.102	288
MiniMax-M2.7	CN	flagship	77	60	71	11.1 s	0.155	0.203	1274

STRETCH cohort

Model	Org	Tier	Acc %	n	Cmpl %	Latency	¢/Q	¢/correct	tok/Q
Claude Opus 4.8	US	frontier	100	24	100	2.3 s	0.512	0.512	179
Qwen3.7-Max	CN	flagship	100	84	100	15.0 s	0.391	0.391	1005
DeepSeek-V4-Pro	CN	flagship	100	84	100	21.6 s	0.095	0.095	1043
Kimi-K2.6	CN	flagship	100	84	100	27.0 s	0.428	0.428	1231
Step-3.7-Flash	CN	flash	99	84	100	11.9 s	0.325	0.329	2807
GLM-5.1	CN	flagship	99	72	86	13.1 s	0.267	0.271	834
MiniMax-M2.7	CN	flagship	96	48	57	20.5 s	0.184	0.192	1512
Seed-2.0-Lite	CN	lite	95	60	71	34.3 s	0.168	0.177	826
Claude Sonnet 4.6	US	frontier	92	24	100	3.0 s	0.245	0.267	142
GPT-5.4-nano	US	nano	81	84	100	3.8 s	0.068	0.084	527
GPT-5.4-mini	US	budget	80	84	100	2.0 s	0.170	0.213	360
Claude Haiku 4.5	US	budget	79	84	100	2.4 s	0.171	0.217	318
Grok 4.3	US	flagship-mid	71	84	100	2.1 s	0.094	0.132	320
Gemini 3.1 Flash-Lite	US	budget	70	84	100	4.4 s	0.253	0.360	1667
ERNIE-4.5-VL	CN	flagship	68	84	100	28.2 s	0.170	0.250	1319
Llama-4-Maverick	US	open-mid	66	84	100	12.5 s	0.033	0.050	515

Download the full dataset. Linked at the top of the online report: a **19-sheet workbook (XLSX)** — overall, model×cohort, six metric pivots, the four cohort sheets, group/country/tier slices, Chinese-only, cost-erosion, and the raw cells — plus a tidy **aggregate CSV** and the **raw cell-level CSV** (one row per answered question: model, problem id, model answer, gold answer, correct, latency, tokens, cost).

Appendix B — study parameters

- **Problems:** AMC 8 / AJHSME only (700 text-only pool: AMC 8 451, AJHSME 249; Math Kangaroo / AoPS in the store excluded), sampled at run time; auto-graded on exact answer-letter match.

- **Tiers:** difficulty bands by AMC problem number — easy (≈#1–10) / medium (≈#6–16) / hard (≈#11–20) / stretch (≈#20–25, the contest's hardest problems; not a harder competition).
- **Harness:** all-at-once, 12 problems/call, full step-by-step working required, 64K output budget, 900 s call deadline, streaming with retry on 429/5xx/empty.
- **Scale:** 28 rounds, 16 models, 4,896 model×problem attempts (4,740 graded; 263 distinct problems); 264–336 questions per cost-matched model, 96 per frontier model.
- **Spend:** ≈ $6.80 metered API + subscription-rate quality judging.
- **Serving:** OpenRouter (all non-Anthropic) + native Anthropic API. Most Chinese models served from Western GPU clouds (DeepInfra / Together / Novita / Parasail / Chutes) — latency reflects model token-economy and provider throughput, not Chinese domestic compute.
- **Date:** 31 May 2026. All prices and model versions as of run time.