

Efficiency Is Not Raw Power: Meet GPT-6-Astra and the DeepSWE Reality

Julien Laurenceau | Published 2026-09-06 | Updated 2026-09-30

<https://pepitedata.com/deepswe-gpt-6-astra-cost-per-task/>

PepiteData | LinkedIn: Julien Laurenceau | X: @Julien_Lau



DeepSWE

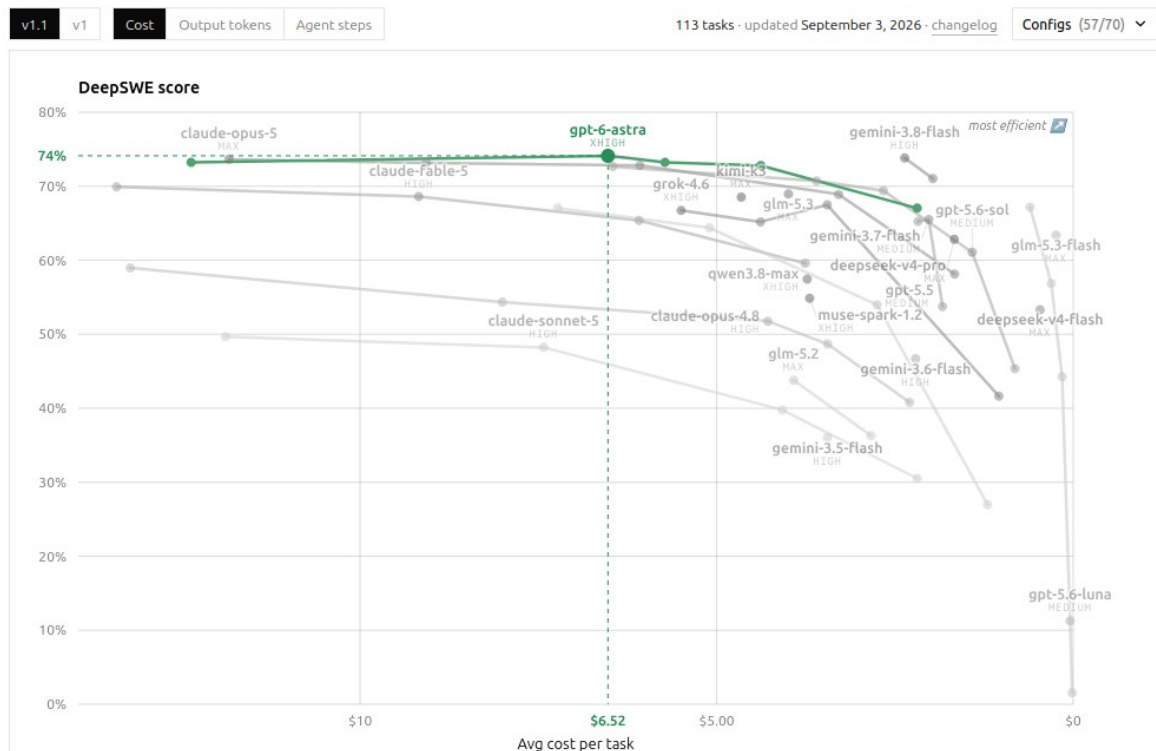
Blog

Run

Data

GitHub

Leaderboard



The marketing machine is running at full speed. Every new model is the next breakthrough, and the industry debates which architecture holds the highest raw intelligence. I spent time inside the DeepSWE dataset, and the picture that emerges is economic: GPT-6-Astra XHIGH sits alone on the cost-performance Pareto front, GLM-5.3-Flash is the sensible production default, and the models separate more sharply on cost per solved task than on peak scores.

GPT-6-Astra XHIGH sits alone on the Pareto front

On DeepSWE, GPT-6-Astra XHIGH is the only configuration that truly sits on the Pareto front of cost versus performance. For every other configuration in the mix, you either pay a premium for slightly less output, or accept a performance hit to save a few cents. In autonomous agent work, where every token and every second is billed, that distinction matters more than the theoretical debate about which model is smartest.

The chart above shows the shape of it: 113 tasks, DeepSWE score on the vertical axis, average cost per task on the horizontal axis. GPT-6-Astra XHIGH sits at \$6.52 per task near the top of the frontier.

The metric that matters: cost per correctly solved task

The real metric is not peak intelligence: it is how much a task, correctly solved, actually costs. A brilliant model that delivers results at ten times the price of its predecessor is not an industrial revolution; it is a luxury play for deep-pocketed enterprises. Production systems need models that deliver value, not spectacle.

My current pick: GLM-5.3-Flash

My personal top pick remains GLM-5.3-Flash. It is not the most spectacular model on the charts, and it does not dominate every benchmark. It offers the best compromise I can find right now between speed, accuracy, and reasonable inference cost. If you are building production systems, this is the model I would reach for first.

The anomaly: a previous-generation model beats a flagship tier

The data also contains a result that defies simple expectations: GPT-5.6-Luna-Max is more efficient than GPT-6-Astra-Low. An older, heavier configuration delivers better value than the entry tier of the newest flagship. Model scaling is not a linear path, and it may be time to revisit assumptions about which architecture fits which workload.

Follow-up: DeepSeek V4.1 Flash scores 98% of Astra at 1.4% of the cost

The open model DeepSeek V4.1 Flash reached 98% of GPT-6 Astra's score at 1.4% of the cost in the OpenDesign Arena benchmark, a result that reads like this post's thesis on another dataset. Per artifact, that is \$0.023 and 5.3 minutes for DeepSeek V4.1 Flash, against \$1.61 and 11.1 minutes for Astra, while Claude Fable 5.1 trailed at \$3.66 and 12.8 minutes.

Three days after I published this analysis, the team behind the [OpenDesign Arena](#) benchmarked DeepSeek V4.1 Flash on everyday design tasks built from user requests. Every model in the comparison except Astra scored lower and cost more. [Full results are in their thread on X.](#)

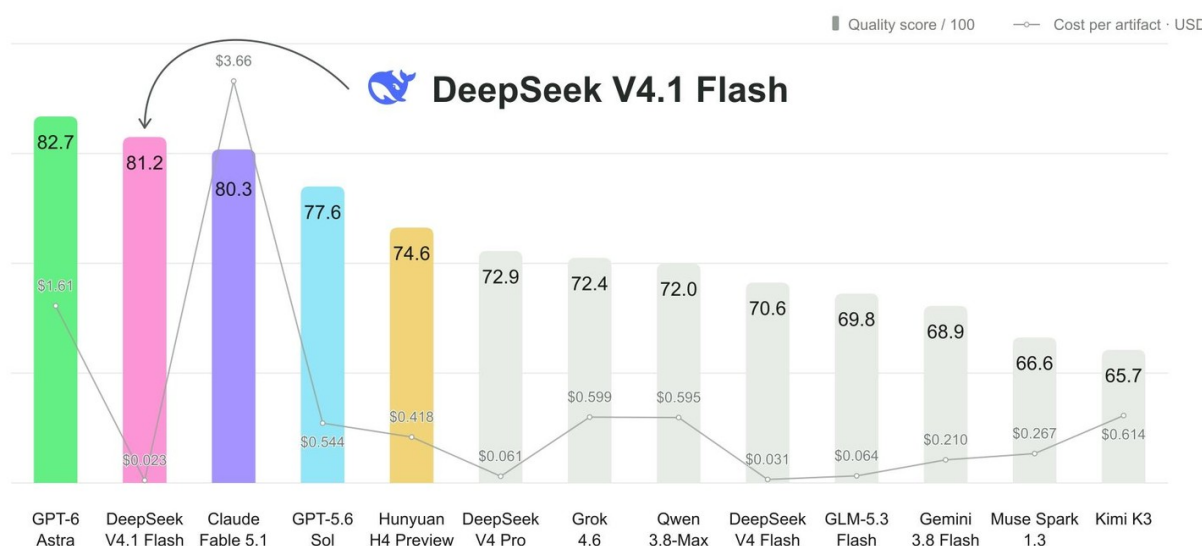
We benchmarked DeepSeek V4.1 Flash ... It reached 98% of GPT-6 Astra's score at 1.4% of the cost on everyday design tasks based on user requests. Every model except Astra scored lower and cost more. Are open models overtaking closed ones?

OpenDesign (@OpenDesignHQ)

, on X

Quality ranking & cost

Source:  OpenDesign



Quality score against cost per artifact across 13 models: DeepSeek V4.1 Flash ranks second on score and first on cost. Chart: OpenDesign.

Two cautions before rewriting a shortlist. This is a design benchmark, not DeepSWE: different tasks, different methodology, no direct comparison between the two leaderboards. And the aggregate hides the detail: DeepSeek V4.1 Flash places ahead of Astra on websites and landing pages but tenth on dashboards and admin panels. The discipline stays the same: read the score measured on a workload that resembles yours, then divide it by the cost.

FAQ

What is DeepSWE?

DeepSWE is a benchmark for autonomous coding agents. The leaderboard behind this article scores model configurations across 113 tasks and reports each success rate next to its average cost per task, so every configuration can be read on both axes at once.

Which configuration offers the best cost-performance on DeepSWE?

On the leaderboard I used, GPT-6-Astra XHIGH is the only configuration that truly sits on the Pareto front of cost versus performance. Every other configuration makes you choose: pay a premium for slightly less output, or accept a performance hit to save budget.

Which model should I pick for production work?

My pick is GLM-5.3-Flash. It does not top every chart, but it offers the best compromise between speed, accuracy, and inference cost for production systems. Treat any leaderboard as a starting point, then validate against your own workload before you commit.

Does DeepSeek V4.1 Flash change the DeepSWE conclusion?

No. The DeepSeek V4.1 Flash figures come from OpenDesign's design benchmark, not from DeepSWE, so the two leaderboards are not directly comparable. It is the same economic pattern on a different

workload: an open model at 98% of the leader's score for 1.4% of its cost per task.

Run the numbers yourself

Run the numbers yourself: the dataset behind this analysis is available in an interactive explorer, so you can run your own experiments and form your own conclusions at deepswe.datacurve.ai. A benchmark rewards what it measures, so generalize from a leaderboard only onto work that resembles what it scored.

One caution before you generalize from any leaderboard: a benchmark rewards what it measures. I opened the [Benchmark Best Practices series with Amdahl's law](#) and the scaling claims it caps.

So, what is driving your current stack: pure intelligence, or the bottom line? The gap between marketing hype and engineering reality keeps widening, and the only way to bridge it is to look at the hard data.

I posted the headline chart on LinkedIn as well, and the discussion is open there: [join the thread](#).

Update 2026-09-29. This post is now part of [AI Infrastructure and SRE: Running Models in Production](#), the field guide to running models in production.

Related posts

- [AI Infrastructure and SRE: Running Models in Production](#)
- [Benchmark Best Practices, Part 1: Amdahl's Law Caps Your Scaling Claim](#) - read a leaderboard claim against the scaling ceiling the benchmark itself imposes.
- [Real-Time AI Streaming Is a Product Trend, Not a Benchmark](#) - how product narratives shape model comparisons.
- [Ox Alpha Review: Test a Free AI Model Before Production](#) - the evaluation gate I would run before any model route reaches production.

And if you want the same cost-versus-performance discipline applied to your own data platform, that is what my [data platform audits](#) are for.

PDF snapshot of the public article. The online version is authoritative; links remain clickable.