The Coding Assistant Breakdown: More Tokens Please

Hands On With GPT 5.5, Opus 4.7, DeepSeek V4, Why Benchmarks Are Bad, and Who's Going To Win

MAX KAN, JORDAN NANOS, SAMUEL KRUSE, AND 4 OTHERS

APR 25, 2026 · PAID

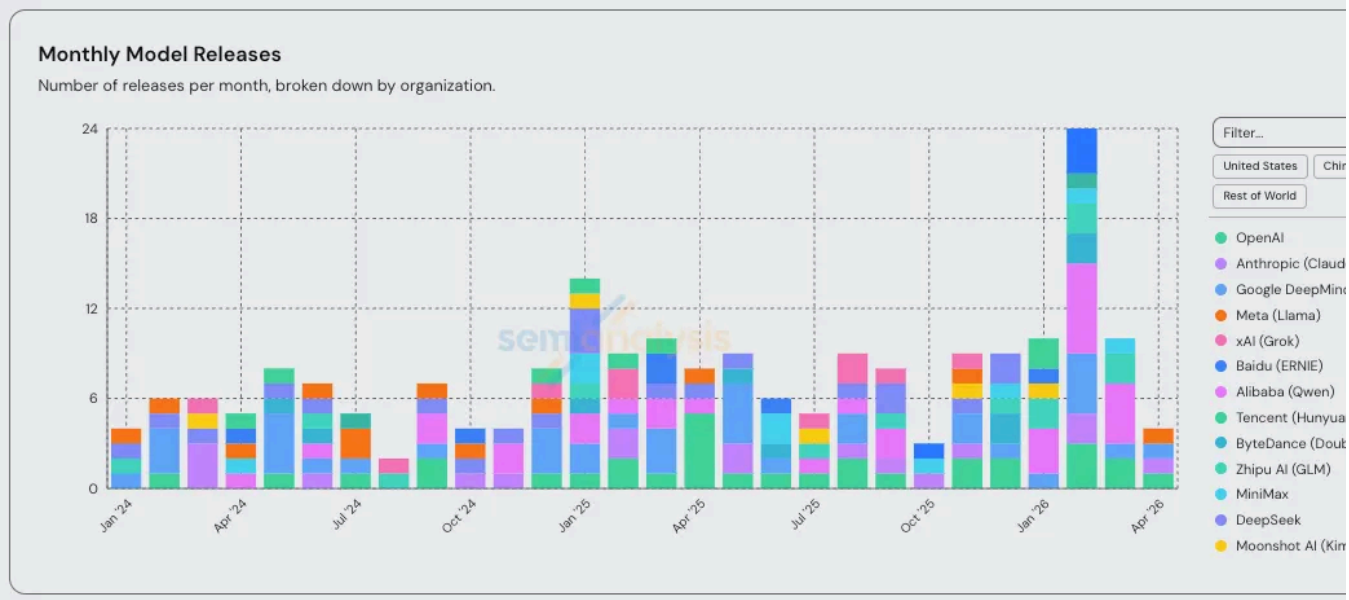
63 8 Share

Since we called out the Claude Code inflection point on February 5th, we have see flurry of model releases. Opus, Mythos, Codex, Gemini, DeepSeek, Kimi, Qwen, G MiniMax, Composer, Muse Spark, and more. Today we will break down all of thes major model releases, explain when you can vs can't trust the benchmarks, and giv our predictions for the future of the agentic coding market.

First we have to highlight GPT-5.5 from OpenAI. In our view, GPT-5.5 is now **materially better** at some tasks than all other models. We believe that GPT-5.5 has arrived at the frontier. This is a huge change from November when Opus 4.5 was released. At that time, and for the 6 months since, OpenAI's coding model was no world class in most metrics, leading to Opus being our daily driver. GPT-5.5 is nov integrated in our daily work.

Meet the Models

There's been at least one major lab releasing a new checkpoint purpose-built for coding every week for the past 3 months. GLM-5.1, Qwen3.6-Plus, Kimi K2.6, Composer 2, and Gemini 3.1 Pro all emphasize "agentic coding," "long-horizon tas or similar capabilities in their headlines. February was a particularly busy month.



Source: SemiAnalysis Tokenomics Dashboard

New checkpoints are cool, but entirely new pre-trains are what really get the peop going. Heading into April, the San Francisco rumor mill was ablaze with talk abou Capybara and Spud. These are codenames for Anthropic and OpenAI's newest pre trains. With the release of GPT-5.5 yesterday, we now have something concrete to discuss.

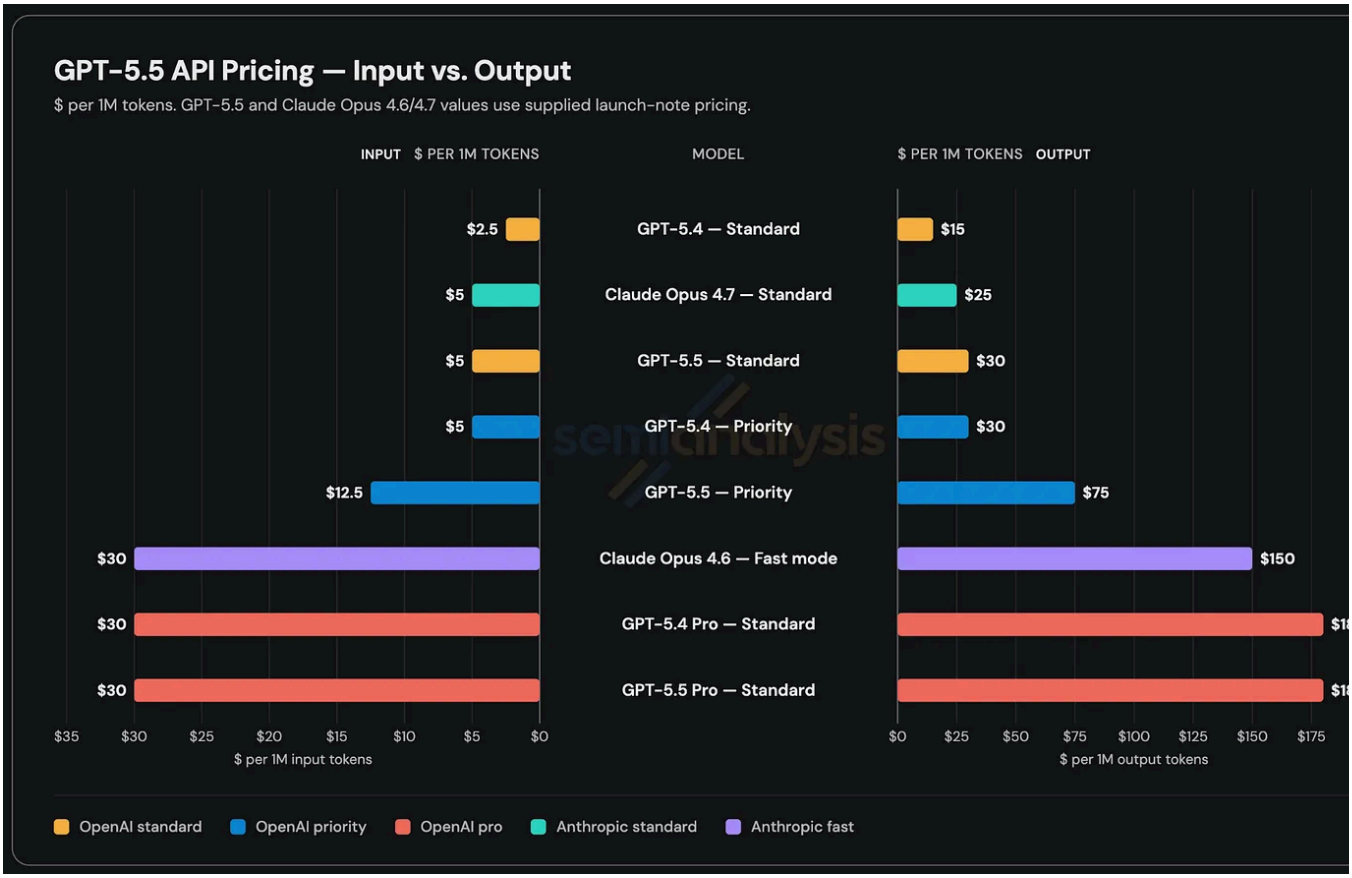
GPT 5.5

GPT-5.5 is the first public release based on "Spud". As OpenAI's first new pre-trai since the failed GPT-4.5, expectations are obviously high. And despite both NVID and OpenAI claiming with precise language that the model was "trained" on a 10(GB200 NVL72 cluster, this "training" is post-training (RL) only. Pre-training is sti Hopper.

OpenAI's flagship model has historically been cheaper than Anthropic's, but at $5 million input tokens and $30 per million output tokens, GPT-5.5's API price will b more expensive than GPT-5.4 and slightly more expensive than Opus 4.7. The API went live this morning after a brief ChatGPT/Codex-only window due to safety concerns. We've been testing the model via Codex and API during an alpha testing period and describe that experience later in this article.

Like all their other models, OpenAI will also be offering a priority tier for GPT-5.5 priced at 2.5x the standard rate. Figuring out how to charge users more money for faster tokens is becoming increasingly important, and it's worth clarifying that pri is totally different from fast mode. Fast mode just makes some vague guarantees li "2.5x faster for 6x the price," whereas priority makes more conservative, concrete S (e.g. > 50 tokens/sec > 99% of the time). Both Anthropic and OpenAI offer fast mod and priority tiers, but we think Opus 4.6 Fast is the only SKU that's gained real traction.

Separately, OpenAI also offers GPT-5.3-Codex-Spark, but that's a totally different model built to run on Cerebras. Specifically, it is a distilled version of GPT-5.3. Th a big difference between offering faster tokens via running smaller batch sizes, changing the reasoning depth, and routing requests to a priority queue without changing the underlying model (priority and fast mode) vs running a dumber, sma model (codex spark).

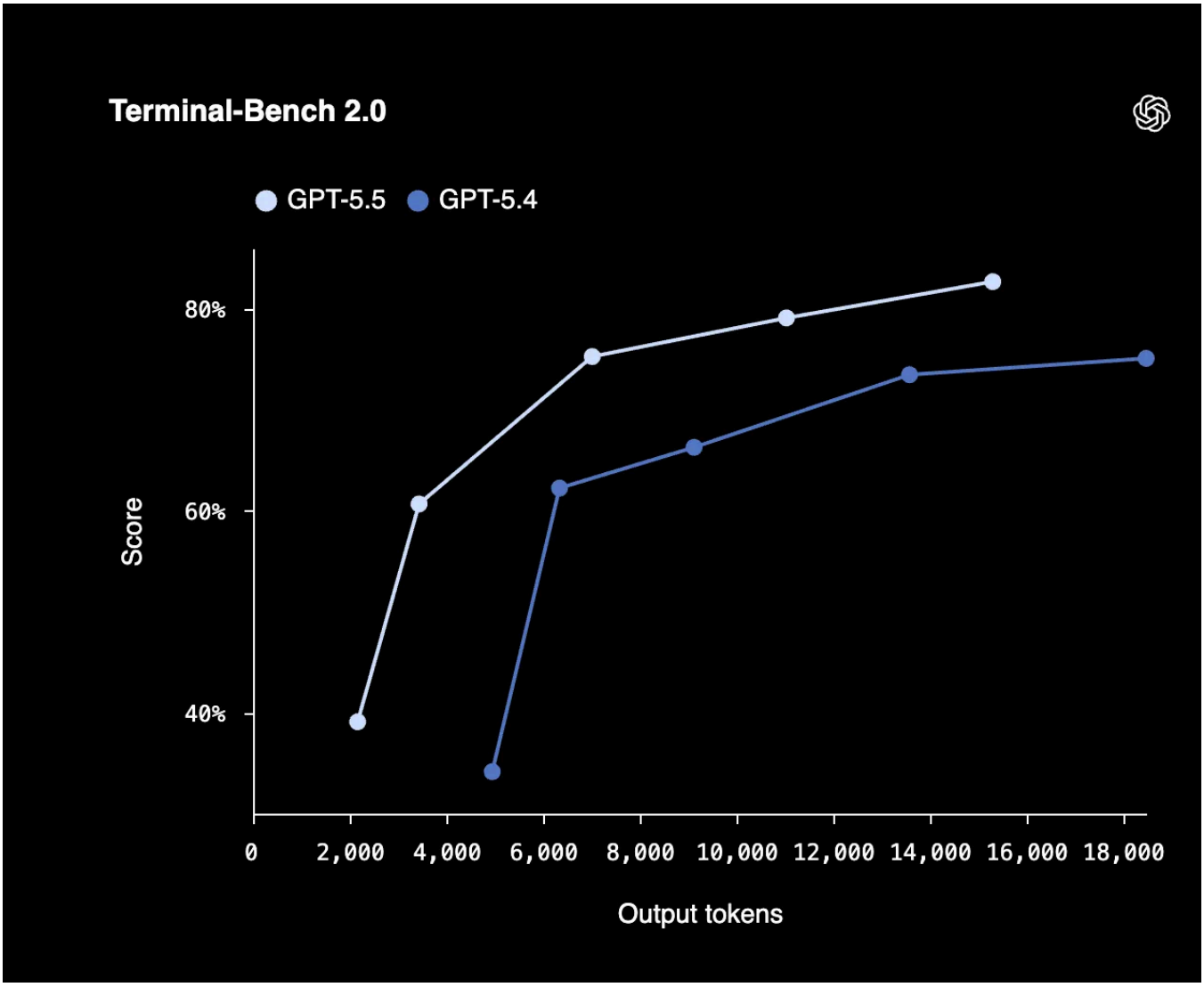


Source: SemiAnalysis

Also released is GPT-5.5 Pro, which is only available via ChatGPT and API. It's m for scientific research or long range reasoning tasks instead of everyday agentic w GPT-5.5 Pro earned SOTA scores on BrowseComp and FrontierMath, and is price the same $30/180 as GPT-5.4 Pro. We expect to see more announcements about GI 5.5 Pro making scientific discoveries soon.

Both the standard and pro models offer different levels of reasoning: xhigh, high, medium, low, and non-reasoning, which is a tradeoff between cost vs capability. A has been clear since the release of strawberry/o1, higher reasoning levels lead to b outputs but require more tokens and users have to wait longer for a response.

Relatedly, OpenAI advertised in their model card that GPT-5.5 scores higher on benchmarks than 5.4 while simultaneously using less tokens. In other words, it's n "token efficient." This is an extremely important concept to understand, and we believe it will become a major talking point this year. As we explained and quantif to Tokenomics model subscribers last week, **cost per task, not cost per token, is t true north star metric that determines model pricing**. Mythos may be 5x more expensive than Opus on a per token basis, but much of that price increase is nullif because Mythos can solve the same problem using fewer tokens. It may also be a f end to end response.



Source: OpenAI

Opus 4.7

This all comes a short week after Anthropic's release of Claude Opus 4.7, a drop-i replacement for Claude Opus 4.6. Opus has been the daily driver for most of SemiAnalysis, and Opus 4.7 is a small improvement. With improved scores on ma benchmarks and predictably good vibes, but not a step change, 4.7 has been reluct adopted by our team members. Why? Fast mode does not exist yet. For the first tir we have found that many of our engineers are willing to sacrifice a bit of quality (b not too much) for faster speed, claiming that the 2.5x faster for 6x the price tradeo lets them hit "flow state".

Source: of our frustration (i.e. Dylan on X)

In practice, the noticeable changes moving from Opus 4.6 → Opus 4.7 have been f features/functionality rather than raw performance. In general, these models have gotten so good that most day-to-day tasks are accomplished successfully, with our engineers' criticisms of a code edit or PR being more about style, approach, architectural decisions, and token efficiency (i.e. speed) rather than success on functional tests. It is increasingly rare for any of these coding models to go haywir and botch a commit completely.

As a result, the noticeable changes in this transition are:

1. High-resolution image support, and a clear increase in RL training objectives include the use of screenshots for frontend styling rather than running tests programmatically via headless browsers and tools like playwright

2. An “xhigh” reasoning effort option that slots in between “high” and “max” on hierarchy of effort (i.e. how much time the model is going to spend reasoning about a task, described earlier)
3. Thinking content is omitted by default. Of course, you still get charged for the tokens, but you have to opt in to see them.
4. Task budgets (in beta, and API only) where the model is given a suggestion on how efficiently to complete a task. If the model is given a task budget that is t restrictive, it can take shortcuts or refuse. This is different from max_tokens, which is a hard restriction on output length
5. Updated token counting, the most critical change when it comes to pricing. 4. uses a new tokenizer, which trades off improved performance via more granul token counting for more total token usage. They admit directly that this will l to increases up to 35% in token usage. Implicitly, this is a 35% increase in pric

On model behavior changes, the biggest thing we have noticed in our testing is ho 4.7 is using fewer tool calls by default, and using reasoning more. The jury is still o on the benefits here, but in general we don’t like it. Anthropic suggests raising the reasoning effort from high to xhigh or max to increase tool usage. And it seems th our users are doing exactly this in order to let the model bring in enough context t successfully complete a complex task or form a complete multi-step plan. Not exa the token efficiency tradeoff claimed in the announcement.

Notably, many people have been accusing Anthropic of intentionally degrading th model on the lead up to the 4.7 release. Anthropic has categorically denied these claims, but multiple engineers at SemiAnalysis independently said that over the la few weeks the changes in 4.6 performance have made them “feel a little schizo”. A course, they were right.

On April 23, a week after the Opus 4.7 release, Anthropic posted a postmortem detailing three bugs that they found in March/April. All three were present for we and affected basically all users of Claude Code. One of the bugs is trivial, two are interesting, and all are real. When the harness is part of the product, the model ge blamed.

We take reports about degradation very seriously. We never intentionally degrade our models, and we were able to immediately confirm that our API and inference layer were unaffected.

After investigation, we identified three different issues:

1. On March 4, we changed Claude Code's default reasoning effort from `high` to `medium` to reduce the very long latency—enough to make the UI appear frozen—some users were seeing in `high` mode. This was the wrong tradeoff. We reverted this change on April 7 after users told us they'd prefer to default to higher intelligence and opt into lower effort for simple tasks. This impacted Sonnet 4.6 and Opus 4.6.
2. On March 26, we shipped a change to clear Claude's older thinking from sessions that had been idle for over an hour, to reduce latency when users resumed those sessions. A bug caused this to keep happening every turn for the rest of the session instead of just once, which made Claude seem forgetful and repetitive. We fixed it on April 10. This affected Sonnet 4.6 and Opus 4.6.
3. On April 16, we added a system prompt instruction to reduce verbosity. In combination with other prompt changes, it hurt coding quality and was reverted on April 20. This impacted Sonnet 4.6, Opus 4.6, and Opus 4.7.

Source: Anthropic Postmortem

Notably the three timelines are March 4 to April 7, March 26 to April 10, and Apri to April 20. This is weeks and weeks of bugs going unnoticed. Bugs that were introduced by Claude, and likely root-caused by Claude. Live by the sword, die by sword.

DeepSeek V4

The long awaited DeepSeek v4 drop is here. DeepSeek took the world by storm las year with its R1 release and since then there have been legitimate questions in the community about whether open source models will commoditize intelligence. For those keeping score at home, DeepSeek crashed the market so hard that CEOs we scrambling to explain Jevons paradox. This seems to have played out quite clearly the 16 months since, with the Great GPU Shortage now upon us.

V4 is an improvement over V3, but it didn't crash the market today. That said, the achievements of DeepSeek should not be discounted. They open-sourced the weig a detailed technical report, and updated libraries such as DeepEP, DeepGEMM, a FlashMLA that are widely used by labs around the world. Ironically, DeepSeek is helping American open source AI stay alive.

This release includes two models: DeepSeek-V4-Pro and DeepSeek-V4-Flash. The former is 1.6T total / 49B active, and the latter is 284B total / 13B active. Pro is a st up from V3, which was 671B total / 37B active, while Flash is a step down. We beli that both these architectures are still meaningfully behind their closed-source counterparts on the frontier in terms of both total and active parameter counts. W detail more about how we model the architectures of leading closed source frontie models in our Tokenomics model.

The core advancement of V4 over V3 is a move from a 128k context window to 1M context. As a result, all of the main technical advancements are focused on long context performance. These include:

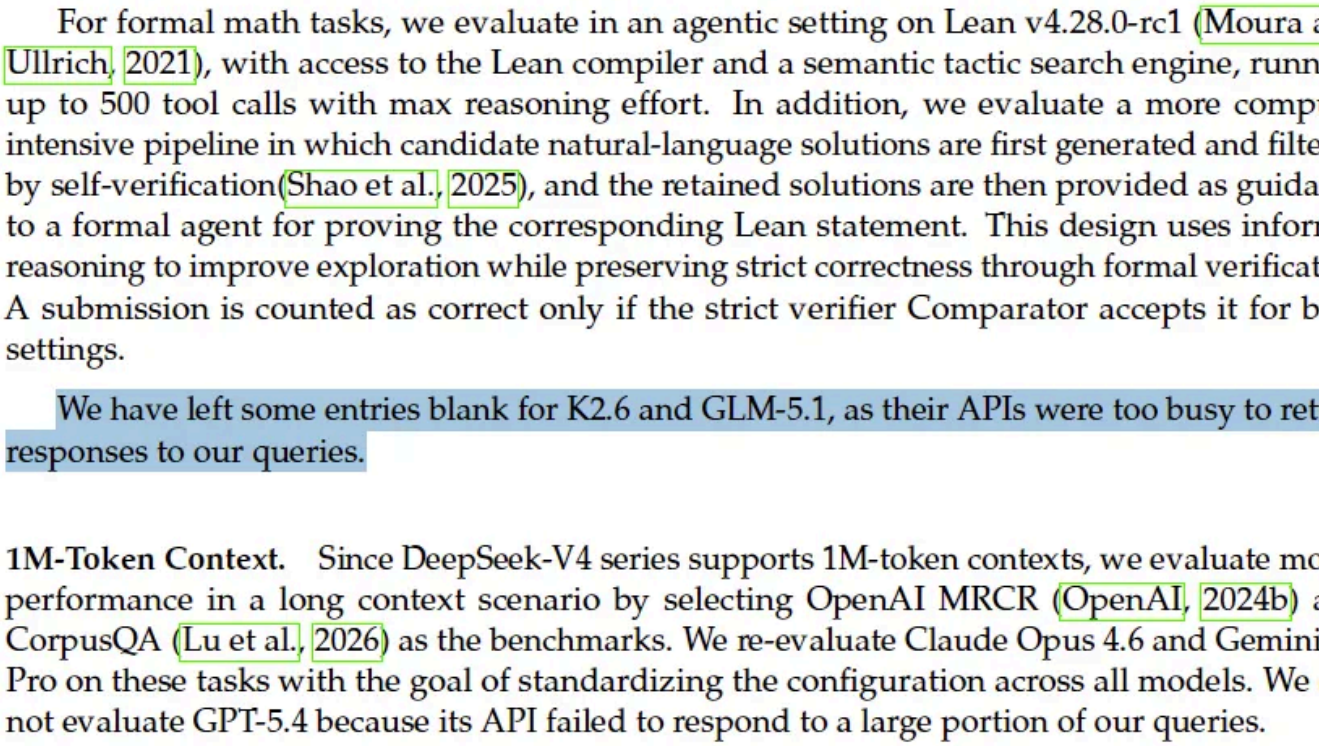
- Compressed Sparse Attention (CSA)
- Heavily Compressed Attention (HCA)
- Manifold-Constrained Hyper-Connections (mHC)

And result in the following claim: "In the one-million-token context setting, DeepSeek-V4-Pro requires only 27% of single-token inference FLOPs and 10% of I cache compared with DeepSeek-V3.2." That's a 90% reduction in KV Cache, way m impactful than Google's TurboQuant paper last month! NAND Flash investors, wa out.

On benchmarks, DeepSeek did not feel that standard benchmarks were good at capturing real-world task capability, so they introduced their own set of agentic benchmarks to measure how V4 compared against other SOTA models: Chinese writing, retrieval augmented search, a suite of white-collar tasks with long horizo and coding. V4 Pro was able to compete with top models on all these tasks but lag behind in key areas. For instance, on especially difficult Chinese writing tasks, Cla

Opus 4.7 still beats DeepSeek V4 Pro. Claude mogs Chinese models in it's own language.

Unfortunately, using public announcements on model performance benchmarks a proxy for real world performance is unreliable. Conflicting incentives cause these to publish certain benchmarks and not others. Like this example, where DeepSeek takes a shot at the Kimi and GLM APIs:



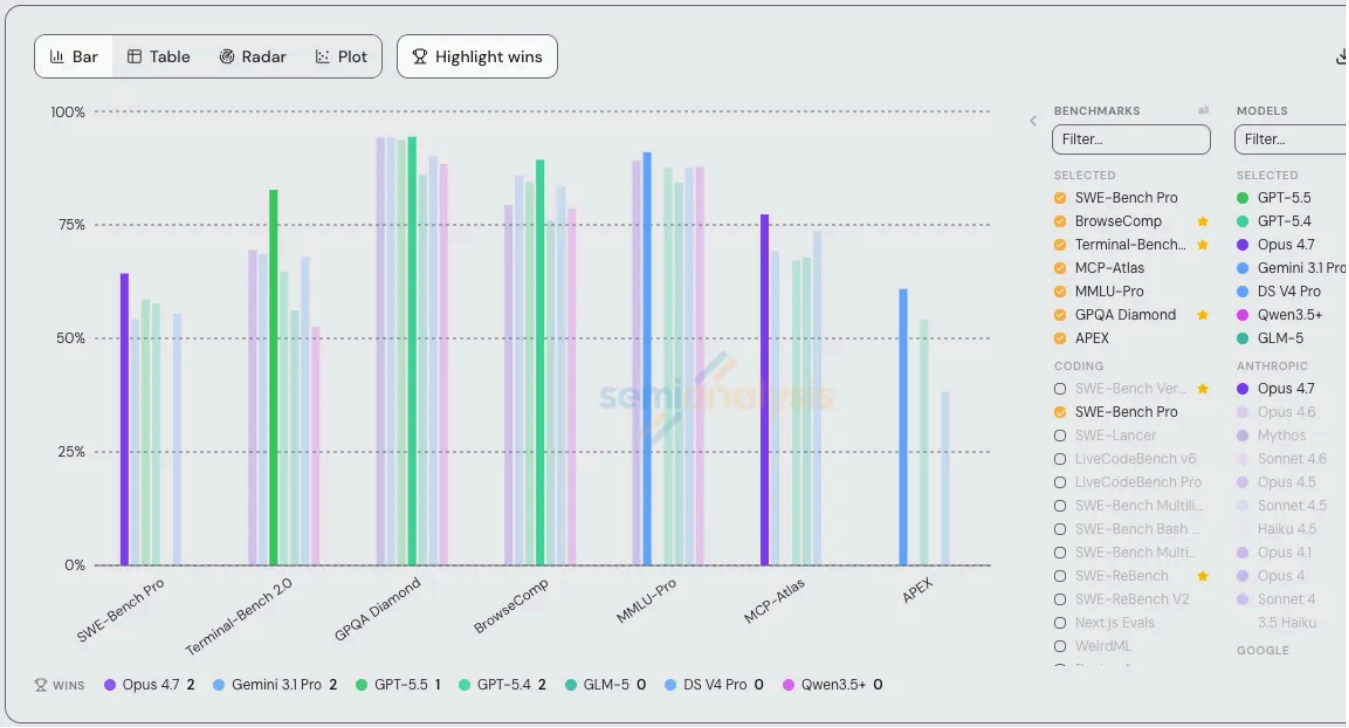
For formal math tasks, we evaluate in an agentic setting on Lean v4.28.0-rc1 (Moura a Ullrich, 2021), with access to the Lean compiler and a semantic tactic search engine, runn up to 500 tool calls with max reasoning effort. In addition, we evaluate a more comp intensive pipeline in which candidate natural-language solutions are first generated and filte by self-verification(Shao et al., 2025), and the retained solutions are then provided as guida to a formal agent for proving the corresponding Lean statement. This design uses infor reasoning to improve exploration while preserving strict correctness through formal verificat A submission is counted as correct only if the strict verifier Comparator accepts it for b settings.

We have left some entries blank for K2.6 and GLM-5.1, as their APIs were too busy to ret responses to our queries.

1M-Token Context. Since DeepSeek-V4 series supports 1M-token contexts, we evaluate mo performance in a long context scenario by selecting OpenAI MRCR (OpenAI, 2024b) a CorpusQA (Lu et al., 2026) as the benchmarks. We re-evaluate Claude Opus 4.6 and Gemini Pro on these tasks with the goal of standardizing the configuration across all models. We not evaluate GPT-5.4 because its API failed to respond to a large portion of our queries.

Source: DeepSeek V4 Technical Report

This is the reason why the SemiAnalysis Tokenomics Dashboard tracks all major model performance claims, pricing, release dates, usage disclosures in an unbiased manner. We also do our own hands-on testing of all the major models. Below is an example of our tracking of meaningful benchmark performance across the major model releases. We will explain later why benchmarks are bad.



Source: Tokenomics Model

Source: Tokenomics Model

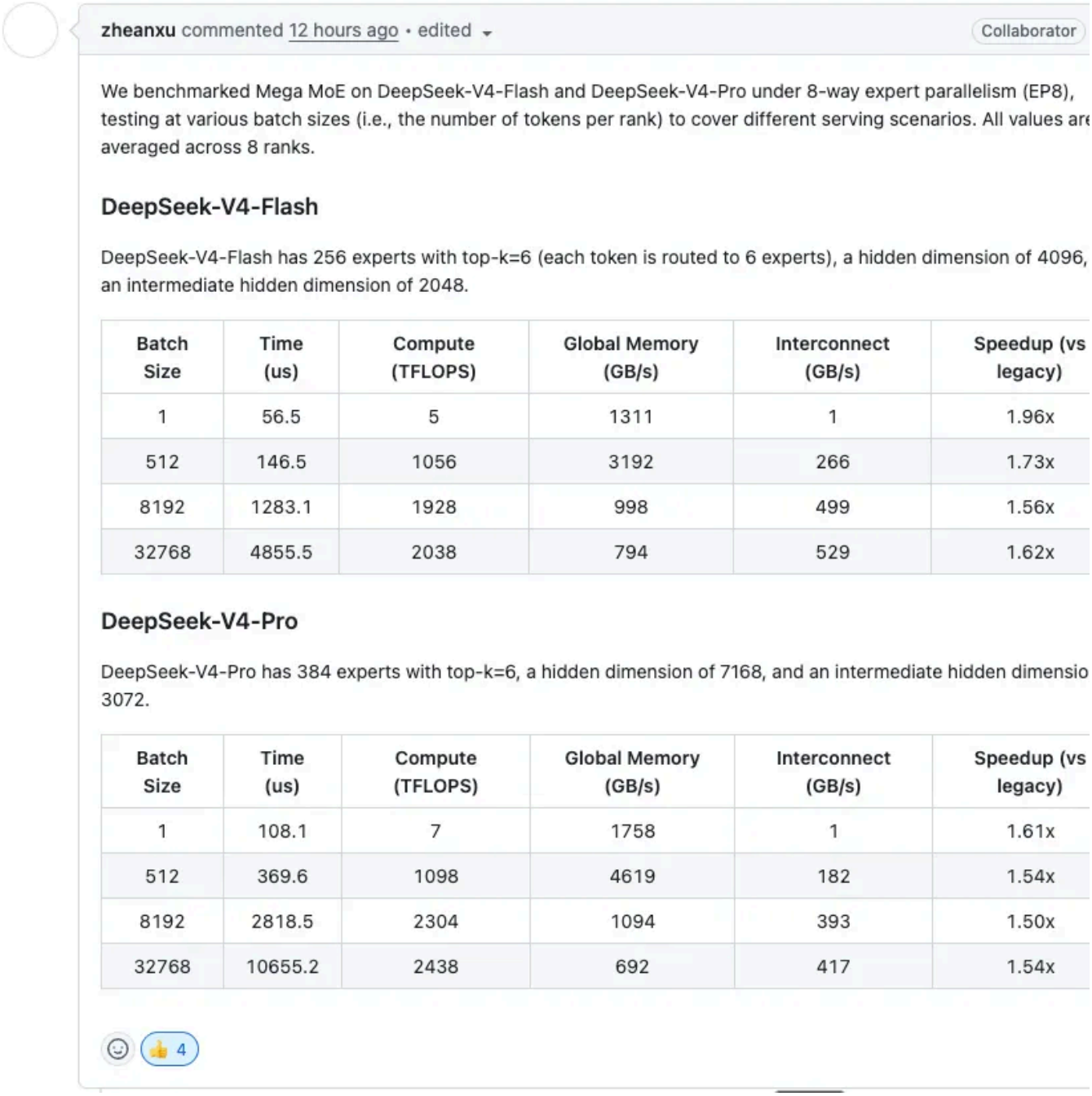
DeepSeek also open sourced a Mega-Kernel inside of DeepGEMM that supports b NVIDIA GPUs and **Huawei Ascend NPUs**. NPU support is claimed, but only the for SM90 (Hopper) and SM100 (Blackwell) GPUs is released publicly. It is likely a g to run a meaningful portion of the future inference traffic on Ascends. It is notabl

however that the parameter size fits just inside the memory domain of an 8x H20 l at FP4.

Performance and Open-Sourced Mega-Kernel. We validated the fine-grained EP scheme on both NVIDIA GPUs and HUAWEI Ascend NPUs platforms. Compared against strong non-fused baselines, it achieves 1.50 ~ 1.73× speedup for general inference workloads, and up to 1.96× for latency-sensitive scenarios such as RL rollouts and high-speed agent serving. We have open-sourced the CUDA-based mega-kernel implementation named **MegaMoE**[2] as a component of DeepGEMM.

Source: DeepSeek V4 Technical Report

Mega MoE performance across various batch sizes is described in a PR:



zheanxu commented 12 hours ago • edited — Collaborator

We benchmarked Mega MoE on DeepSeek-V4-Flash and DeepSeek-V4-Pro under 8-way expert parallelism (EP8), testing at various batch sizes (i.e., the number of tokens per rank) to cover different serving scenarios. All values are averaged across 8 ranks.

DeepSeek-V4-Flash

DeepSeek-V4-Flash has 256 experts with top-k=6 (each token is routed to 6 experts), a hidden dimension of 4096, an intermediate hidden dimension of 2048.

Batch Size	Time (us)	Compute (TFLOPS)	Global Memory (GB/s)	Interconnect (GB/s)	Speedup (vs legacy)
1	56.5	5	1311	1	1.96x
512	146.5	1056	3192	266	1.73x
8192	1283.1	1928	998	499	1.56x
32768	4855.5	2038	794	529	1.62x

DeepSeek-V4-Pro

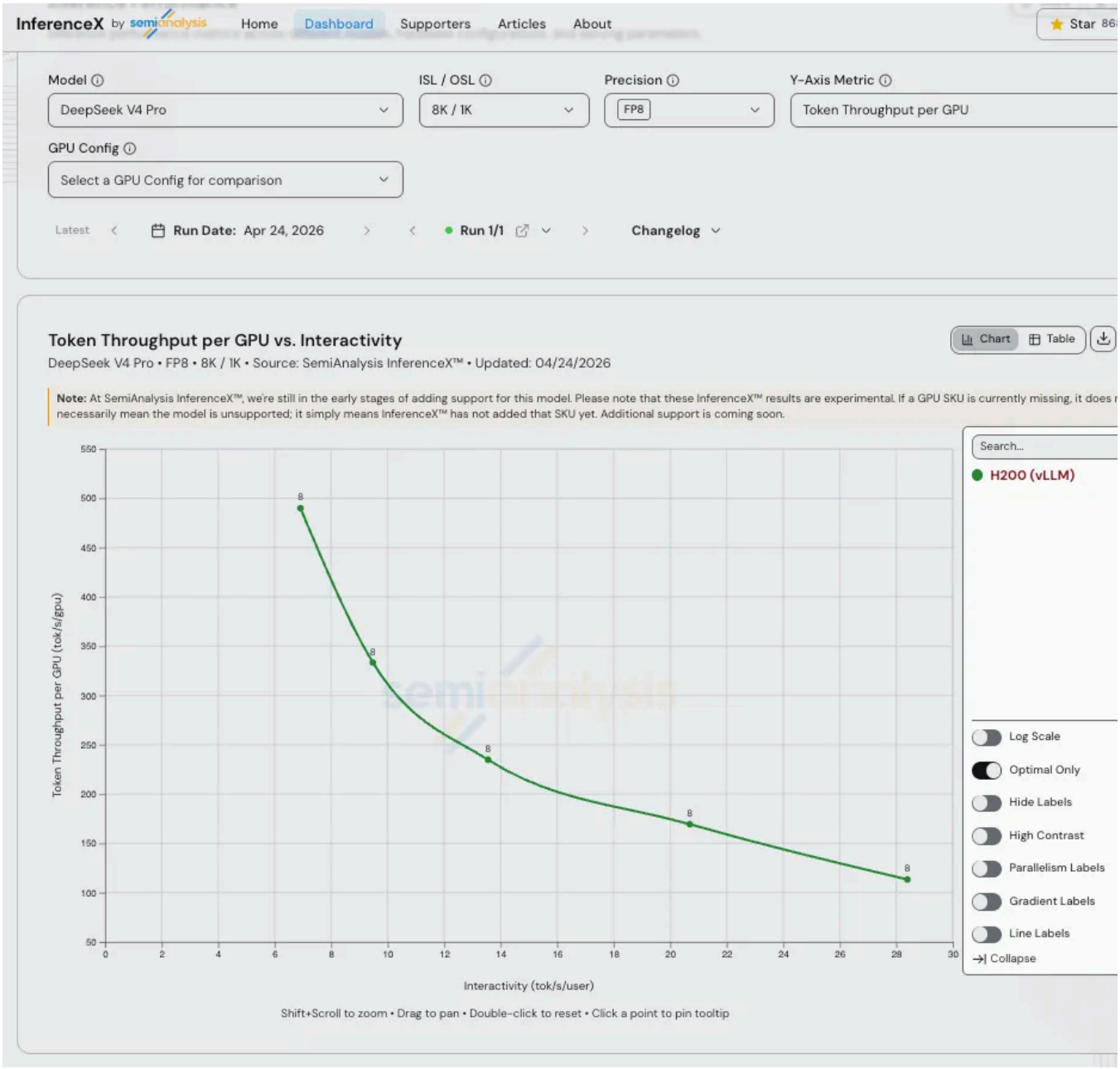
DeepSeek-V4-Pro has 384 experts with top-k=6, a hidden dimension of 7168, and an intermediate hidden dimensio 3072.

Batch Size	Time (us)	Compute (TFLOPS)	Global Memory (GB/s)	Interconnect (GB/s)	Speedup (vs legacy)
1	108.1	7	1758	1	1.61x
512	369.6	1098	4619	182	1.54x
8192	2818.5	2304	1094	393	1.50x
32768	10655.2	2438	692	417	1.54x

4

Source: DeepGEMM repo

Of course, the key contribution of DeepSeek V4 is that it is open source. Thanks t all nighter, our InferenceX team, collaborating with 10x engineers from vLLM/Inf and NVIDIA, have published day-zero support on our H200 cluster. Support for Blackwell and AMD GPUs using vLLM, SGLang and TRT-LLM with Dynamo is a work in progress.



Source: inferencex.com

Interestingly, day-zero support on H200 at FP8 [1] performance of this model hits ~1 tok/sec throughput per GPU at 20 tok/sec interactivity on 8k in 1k out. For referen V3 hits ~1.3k to 2.3k tok/sec of throughput per GPU at 20 tok/sec interactivity on 8 1k out. This is a new model and we expect meaningful optimization in the coming weeks. Watch inferencex.com for real time improvements.

Model	#Total Params	#Activated Params	Context Length	Precision	Download
DeepSeek-V4-Flash-Base	284B	13B	1M	FP8 Mixed	HuggingFace ModelScope
DeepSeek-V4-Flash	284B	13B	1M	FP4 + FP8 Mixed*	HuggingFace ModelScope
DeepSeek-V4-Pro-Base	1.6T	49B	1M	FP8 Mixed	HuggingFace ModelScope
DeepSeek-V4-Pro	1.6T	49B	1M	FP4 + FP8 Mixed*	HuggingFace ModelScope

**FP4 + FP8 Mixed: MoE expert parameters use FP4 precision; most other parameters use FP8.*

Source: DeepSeek V4 model card on Huggingface

Overall, DeepSeek is an exceptional engineering release, and is right behind the S frontier. It will be the lowest cost alternative to closed source models, but it's capabilities are not at the leading edge. SemiAnalysis's workflows likely will not b cannibalized by DeepSeek.

VIBEZ: Our Impressions of GPT-5.5 vs Opus 4

SemiAnalysis is famous (infamous?) for shilling Claude, and we've been testing GP 5.5 as part of an alpha program with OpenAI the past few weeks.

We think GPT-5.5 is a significant improvement within Codex specifically. Previou ~all our engineers used Claude exclusively, and use of ChatGPT models for coding restricted to wrappers like Cursor. Now, most of our engineers switch between Co and Claude models depending on the task and IDE preference. Here are some quo

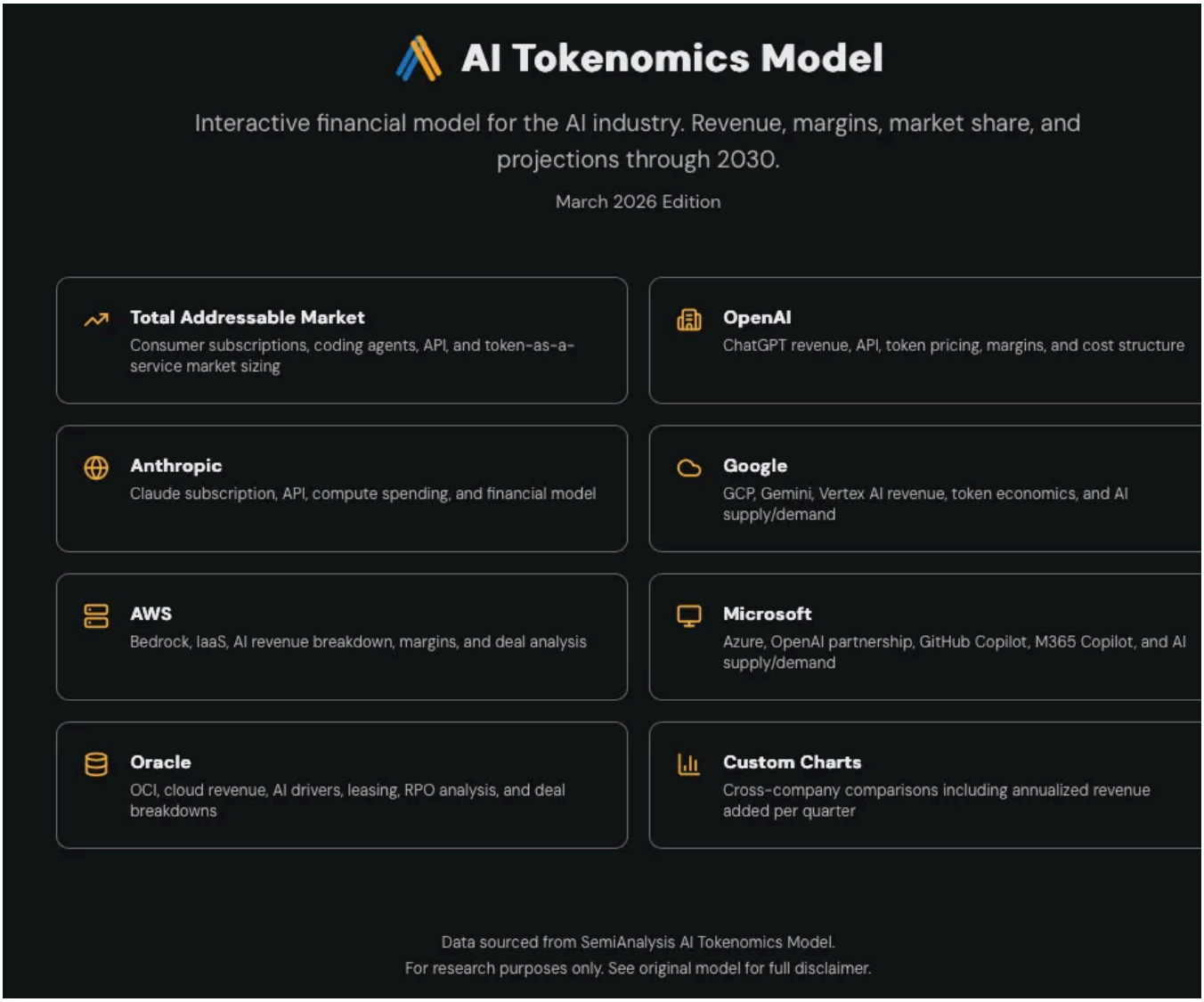
> *"What I have really appreciated about Codex recently is how it pulls in a lot of context before making changes to code. Not like just a structural change, but a change that act requires non trivial 'thinking'. 4.7 often feels like it just does a quick Explore and then*

> *#yolos changes whereas codex pulls in a shit ton of more granular context from the inte + codebase and then makes a directed effort at the ask"*
>
> *"Currently I use Codex for reviewing PRs/bug hunting, explaining existing code, and creating/revising documentation. Its better at understanding code structure and reason about it."*

However, it's not all positive for OpenAI. Some of our other engineers complained Codex is still worse at inferring your true intent than Claude Code. Humans natur give terse and not particularly well thought out instructions when prompting codi agents, and Codex often listens too literally.

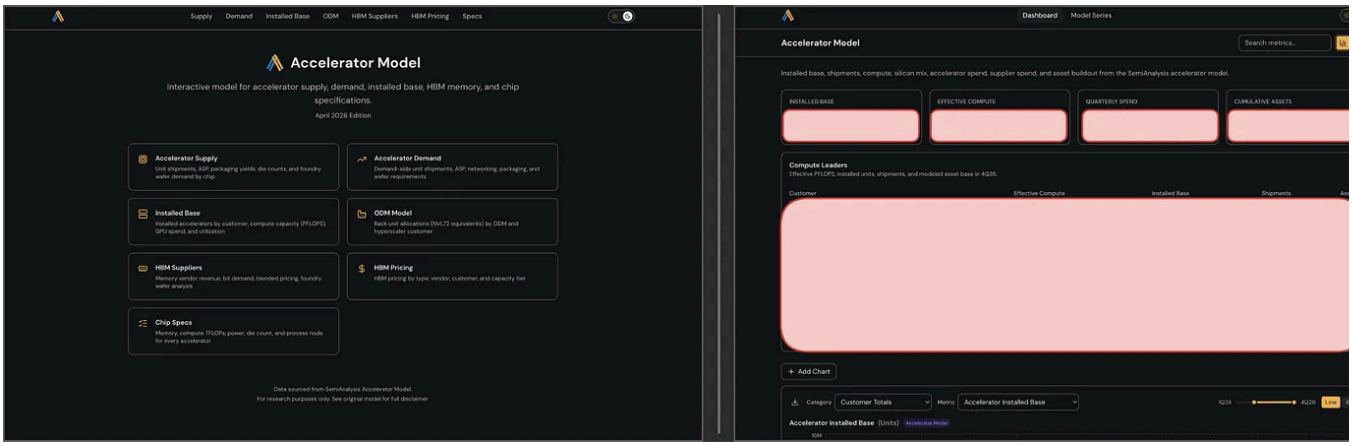
Relatedly, another engineer commented that GPT-5.5 feels too conservative when comes to actually making code changes. Yes, this improves token efficiency, but it comes at the cost of correctness. A similar tradeoff happened from 4.6 → 4.7 as we described previously. Seeing the words "narrow fix" in the output is now a signal t double check the model's work.

Here's a concrete example that illustrates our overall impression on the strengths weaknesses of Codex vs Claude Code well. We asked both Opus 4.6 and GPT-5.5 t make a new dashboard for our accelerator model and gave it the current



Source: SemiAnalysis

Opus 4.6 made an identical looking homepage, whereas Codex ignored it entirely.



Source: SemiAnalysis

If we specifically asked Codex to copy the homepage in the prompt, we're sure it would've done so, but it was unable to infer this intent itself.

With that said, the actual data Codex included in the dashboard was much more accurate than Claude (though to be clear neither was perfect on the first pass). Thi implies stronger reasoning about the data structures and relationships with a relatively complex excel file on the part of Codex. Meanwhile, many of Claude's numbers were straight up hallucinated and it made mistakes like including Nvidia GPUs in TPU charts. This tracks with our overall impression that Codex is "smart and better at doing complex reasoning to solve harder, more narrowly scoped task whereas Claude is better for more open ended, greenfield problems.

It's for these reasons that some of our engineers have settled on the following workflow:

1. Start off with Claude to create an initial plan/scaffolding for new applications features, and the first implementation/POC step.
2. Switch to Codex to actually solve the problem or fix bugs

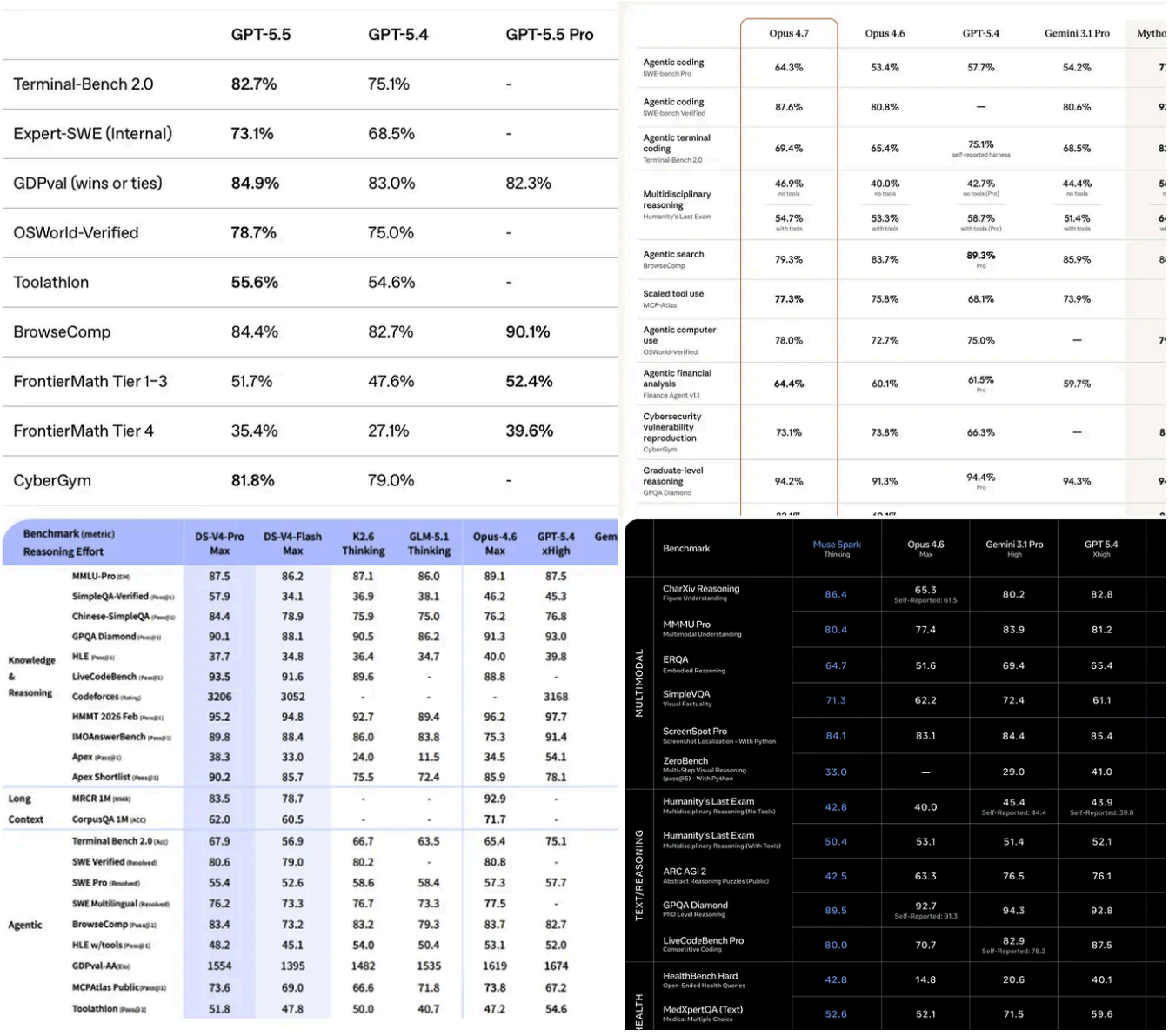
Importantly, before the GPT-5.5 release, ~all of SemiAnalysis used Claude Code fo both of these steps. Our use of ChatGPT models had become restricted to Deep Research on the webapp and wrappers like Cursor Bugbot.

Critically, features in the plugins/CLIs are holding Codex back. Many of our engin prefer fast mode with 1M context, use remote control/sandbox plugins to take sess from laptop to phone and back, and upload images/screenshots during a conversat All of this is possible with the Claude Code CLI, VSCode Plugin, web app and mo app. But none of it is currently possible with the Codex CLI, VSCode Plugin, desk app, web app and mobile app.

Even if GPT-5.5 is a better model, OpenAI needs to ship features at a faster pace i order to catch up with Anthropic and increase adoption.

Benchmarks are bad but we need to keep using them anyways

The one thing that is always prominently featured in every new model announcem is a table comparing performance on various benchmarks.



	GPT-5.5	GPT-5.4	GPT-5.5 Pro
Terminal-Bench 2.0	**82.7%**	75.1%	-
Expert-SWE (Internal)	**73.1%**	68.5%	-
GDPval (wins or ties)	**84.9%**	83.0%	82.3%
OSWorld-Verified	**78.7%**	75.0%	-
Toolathlon	**55.6%**	54.6%	-
BrowseComp	84.4%	82.7%	**90.1%**
FrontierMath Tier 1–3	51.7%	47.6%	**52.4%**
FrontierMath Tier 4	35.4%	27.1%	**39.6%**
CyberGym	**81.8%**	79.0%	-

	Opus 4.7	Opus 4.6	GPT-5.4	Gemini 3.1 Pro	Mytho
Agentic coding SWE-bench Pro	64.3%	53.4%	57.7%	54.2%	7
Agentic coding SWE-bench Verified	87.6%	80.8%	—	80.6%	9
Agentic terminal coding Terminal-Bench 2.0	69.4%	65.4%	75.1% self-reported harness	68.5%	8
Multidisciplinary reasoning Humanity's Last Exam	46.9% no tools 54.7% with tools	40.0% no tools 53.3% with tools	42.7% no tools (Pro) 58.7% with tools (Pro)	44.4% no tools 51.4% with tools	5 6
Agentic search BrowseComp	79.3%	83.7%	**89.3%** Pro	85.9%	8
Scaled tool use MCP-Atlas	**77.3%**	75.8%	68.1%	73.9%	
Agentic computer use OSWorld-Verified	78.0%	72.7%	75.0%	—	7
Agentic financial analysis Finance Agent v1.1	**64.4%**	60.1%	61.5% Pro	59.7%	
Cybersecurity vulnerability reproduction CyberGym	73.1%	73.8%	66.3%	—	8
Graduate-level reasoning GPQA Diamond	94.2%	91.3%	94.4% Pro	94.3%	9

Benchmark (metric) Reasoning Effort		DS-V4-Pro Max	DS-V4-Flash Max	K2.6 Thinking	GLM-5.1 Thinking	Opus-4.6 Max	GPT-5.4 xHigh	Gem
Knowledge & Reasoning	MMLU-Pro (EM)	87.5	86.2	87.1	86.0	89.1	87.5	
	SimpleQA-Verified (Pass@1)	57.9	34.1	36.9	38.1	46.2	45.3	
	Chinese-SimpleQA (Pass@1)	84.4	78.9	75.9	75.0	76.2	76.8	
	GPQA Diamond (Pass@1)	90.1	88.1	90.5	86.2	91.3	93.0	
	HLE (Pass@1)	37.7	34.8	36.4	34.7	40.0	39.8	
	LiveCodeBench (Pass@1)	93.5	91.6	89.6	-	88.8	-	
	Codeforces (Rating)	3206	3052	-	-	-	3168	
	HMMT 2026 Feb (Pass@1)	95.2	94.8	92.7	89.4	96.2	97.7	
	IMOAnswerBench (Pass@1)	89.8	88.4	86.0	83.8	75.3	91.4	
	Apex (Pass@1)	38.3	33.0	24.0	11.5	34.5	54.1	
	Apex Shortlist (Pass@1)	90.2	85.7	75.5	72.4	85.9	78.1	
Long Context	MRCR 1M (MMR)	83.5	78.7	-	-	92.9	-	
	CorpusQA 1M (ACC)	62.0	60.5	-	-	71.7	-	
Agentic	Terminal Bench 2.0 (Acc)	67.9	56.9	66.7	63.5	65.4	75.1	
	SWE Verified (Resolved)	80.6	79.0	80.2	-	80.8	-	
	SWE Pro (Resolved)	55.4	52.6	58.6	58.4	57.3	57.7	
	SWE Multilingual (Resolved)	76.2	73.3	76.7	73.3	77.5	-	
	BrowseComp (Pass@1)	83.4	73.2	83.2	79.3	83.7	82.7	
	HLE w/tools (Pass@1)	48.2	45.1	54.0	50.4	53.1	52.0	
	GDPval-AA (Elo)	1554	1395	1482	1535	1619	1674	
	MCPAtlas Public (Pass@1)	73.6	69.0	66.6	71.8	73.8	67.2	
	Toolathlon (Pass@1)	51.8	47.8	50.0	40.7	47.2	54.6	

	Benchmark	Muse Spark Thinking	Opus 4.6 Max	Gemini 3.1 Pro High	GPT 5.4 Xhigh
MULTIMODAL	CharXiv Reasoning Figure Understanding	86.4	65.3 Self-Reported: 61.5	80.2	82.8
	MMMU Pro Multimodal Understanding	80.4	77.4	83.9	81.2
	ERQA Embodied Reasoning	64.7	51.6	69.4	65.4
	SimpleVQA Visual Factuality	71.3	62.2	72.4	61.1
	ScreenSpot Pro Screenshot Localization - With Python	84.1	83.1	84.4	85.4
	ZeroBench Multi-Step Visual Reasoning (pass@5) - With Python	33.0	—	29.0	41.0
TEXT/REASONING	Humanity's Last Exam Multidisciplinary Reasoning (No Tools)	42.8	40.0	45.4 Self-Reported: 44.4	43.9 Self-Reported: 39.8
	Humanity's Last Exam Multidisciplinary Reasoning (With Tools)	50.4	53.1	51.4	52.1
	ARC AGI 2 Abstract Reasoning Puzzles (Public)	42.5	63.3	76.5	76.1
	GPQA Diamond PhD Level Reasoning	89.5	92.7 Self-Reported: 91.3	94.3	92.8
	LiveCodeBench Pro Competitive Coding	80.0	70.7	82.9 Self-Reported: 78.2	87.5
HEALTH	HealthBench Hard Open-Ended Health Queries	42.8	14.8	20.6	40.1
	MedXpertQA (Text) Medical Multiple Choice	52.6	52.1	71.5	59.6

Source: every release, man

It's very tempting to be able to point to a small set of numbers in order to prove th "objective" superiority of your new model release, but many within the AI commu have long lamented that benchmarks are no longer a useful proxy for real-world ut We tend to agree with this point of view. There's a big difference between claimin

measure a model's coding/finance/reasoning abilities vs actually doing so in any meaningful capacity.

That being said, we expect all the labs to continue highlighting improved benchm performance for all future model releases, and the following section will help you separate the signal from the noise.

Anatomy of a benchmark

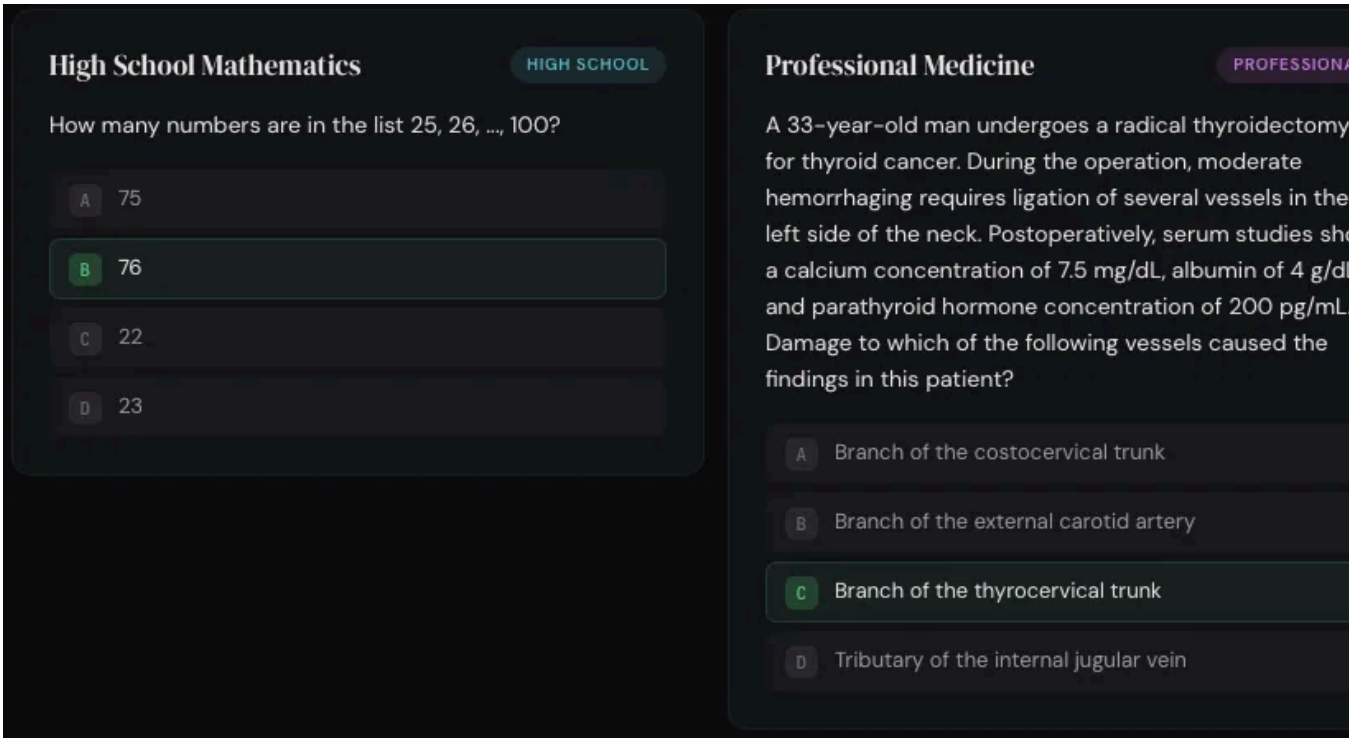
Each benchmark consists of 3 things

1. **Tasks**: what the model is actually asked to do
2. **The evaluation method**: how the model is actually scored
3. **A harness**: what tools, instructions, interface, etc the model is given to solve t task

Really understanding the first two is how you determine if a benchmark is any goc not. To illustrate, we'll walk through some famous benchmarks below in rough chronological order. This will also give you a sense of how benchmarks have chan over time.

MMLU and multiple choice/simple answer benchmarks

Released by academic researchers in 2020, Measuring Massive Multitask Languag Understanding (MMLU) is a set of 15,908 multiple choice questions covering 57 subjects. These questions were manually collected by university students from onl sources like standardized tests and college exams/problem sets. All of them have exactly 4 choices and are publicly available, but they range in difficulty from "elementary" to "advanced professional".



Example MMLU questions. Source: MMLU

MMLU has a minimal harness that essentially just formats the question into a pro Tools like web search are not included. **The multiple choice format is crucial bec it makes grading trivial–just check if the model outputted the right letter.**

MMLU was effectively solved (aka "saturated") by GPT-4 in March 2023 when it scored 86.4%. In practice, the true max score for benchmarks is usually lower than 100% because some of the tasks are ambiguous, poorly worded, or just straight up incorrect. This paper estimates that 6.49% of MMLU questions contain errors for example.

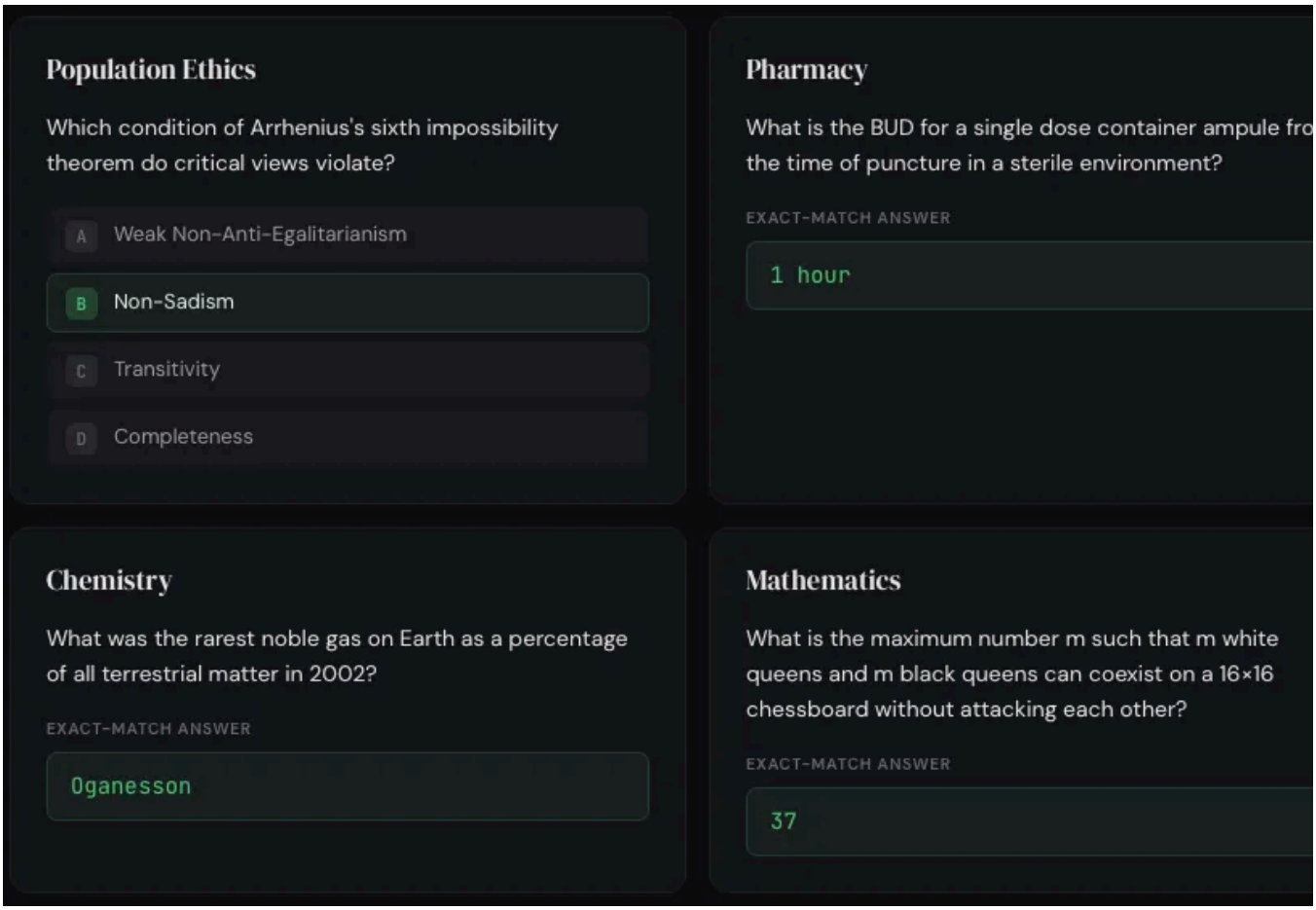
Other benchmarks from the same era include

- GSM8K: Multi-step math problems created by contractors with STEM degree make evaluation simple, all answers are a single number.
- HellaSwag: Multiple choice questions that ask the AI to predict the most likel continuation of an everyday scenario. Tasks are sourced from video captions a WikiHow articles.
- MMMU: The same thing as MMLU except the questions also include images, the model needs vision. The third M stands for "multimodal".

- GPQA: "Google-proof" multiple choice science questions created by 61 PhD-
  contractors.

As each of these became saturated, their creators made harder versions (e.g. MML
Pro, MMMU-Pro, GPQA-Diamond). Tactics include filtering out easy questions fr
the previous version, using an LLM to up the choices from 4 to 10, paying your
contractors to make harder questions, etc.

The most relevant simple answer benchmark today is Humanity's Last Exam (HLE
Released by Scale AI in January 2025, they sourced 1000+ experts from around the
world to create 2500 questions on everything from algebraic geometry to classical
ballet. 80% of the questions require an exact-match short answer and 20% are mult
choice. For the harness, you can choose to run the model with or without tools (e.g
web search and code execution).



Example HLE questions. Source: Scale AI

These questions obviously aren't representative of real-world LLM usage and are a riddled with issues. For example, one study found that 30% of HLE chemistry/biol questions had answers that directly conflicted with peer-reviewed literature.

However, the labs still absolutely hillclimb all of these benchmarks during the RL stage of training. Google, for example, had a 9 figure budget in 2025 specifically fo HLE style STEM questions, which they paid to data vendors like Mercor, Surge, ar Handshake. It's no coincidence that Gemini 3 Pro was a step-change improvemen the benchmark.

Benchmark	Description		Gemini 3 Pro	Gemini 2.5 Pro	Claude Sonnet 4.5	GPT
Humanity's Last Exam	Academic reasoning	No tools	**37.5%**	21.6%	13.7%	26.
		With search and code execution	**45.8%**	—	—	—

Source: Google

The generous explanation for why labs care about things like HLE is that the knowledge gained from solving esoteric multiple choice questions will transfer to other use cases. The cynical explanation is that corporate VPs want to be able to p to a single number to prove that they're doing their job, and Scale was good at marketing HLE to win sufficient mind share.

SWE-bench and coding benchmarks

Coding is the most important AI capability, and SWE-bench (released in 2023) was first big coding benchmark.

The tasks were automatically scraped from 12 **Python** repos, including django, scikit-learn, and seaborn. They used the following 3 step filtering process:

1. Start with all ~93k merged PRs for all 12 repos
2. Reduce to ~11k that were linked to a GitHub issue and introduced new tests
3. Keep just the 2294 PRs where at least one of the new tests fail when applied to commit immediately before said PR

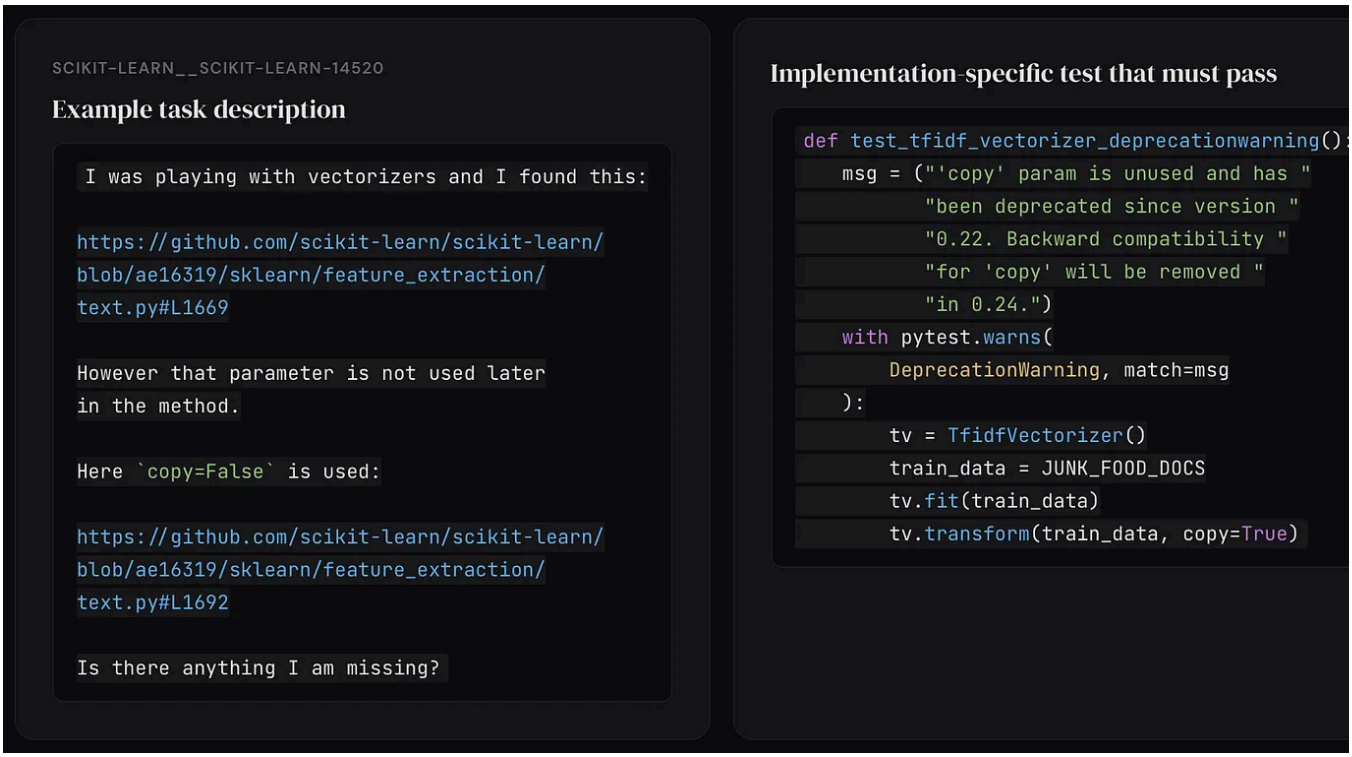
In other words, the GitHub issue is the task, and the PR that resolved said issue is proof that the task is possible. The eval is all the old tests in the repo plus the new

tests included in the PR. The AI is successful if none of the old tests break (pass-to pass) AND all the new tests pass (fail-to-pass). Importantly, the model is not allow see any of the new tests while attempting the task. For the harness, the model is al to inspect the codebase, but it can't actually run any code.

It is worth emphasizing that there was **no human verification** at any step in the ta creation process. GitHub issues are often ambiguous and poorly specified. Furthermore, the tests devs include in their PRs are typically noncomprehensive a scoped to particular implementation details. This causes two big issues

1. If the problem statement allows for multiple solutions, but your tests are scop a single correct solution, then some correct answers will get wrongly rejected
2. If your tests are noncomprehensive, then you will incorrectly pass the AI even only completes a subset of the requirements

In short, many of the SWE-bench tasks were straight up broken. For example, one required the AI to perfectly match a 19 word error message in the eval despite not mentioning it at all in the problem description.



Example SWE-bench task description (left) with an unfair test (right). Source: SWE-bench

OpenAI attempted to solve these issues by releasing SWE-bench verified in Augu 2024. They hired 93 python devs to **manually review** all the task descriptions and e for ambiguity/unfairness. After filtering out all the problematic ones, the original problems were reduced to 500 "verified" tasks. OpenAI also added a bash tool to t harness so the AI could execute code–making the benchmark more agentic–and improved infra reliability by packaging each task as a Docker container.

In February 2026, OpenAI announced that they would no longer report results on SWE-bench verified for two reasons:

1. Of the 138 problems consistently failed by o3, over half *still* had unfair evals th were scoped to specific implementation details not mentioned in the task description OR extra tests that checked for functionality not mentioned in the task description. In other words, the "verified" subset still wasn't very good
2. Because all the PRs are part of popular open-source repos that are definitely included in every model's training data, they found evidence that GPT-5.2, Op 4.5, and Gemini 3 Flash had all memorized some of the answers (aka "contamination").

Instead, they recommended model makers report SWE-bench pro results instead. SWE-bench pro is another Scale creation. The main difference (besides making th tasks harder) is that they used public repos with less permissive licenses and priva repos to avoid contamination. They also hired contractors to write evals and probl descriptions for the commits, instead of relying purely on GitHub issues and preexisting PRs. These are all good steps, but they definitely don't fully solve eithe problem identified with SWE-bench verified. As you've probably already figured o by now, no benchmark is perfect.

SWE-bench pro and verified are both still commonly reported in model release ca today. Other popular coding benchmarks include

- SWE-bench multilingual: Basically SWE-bench verified but with 9 languages instead of just Python

- Terminal-bench: Tasks and evals are both crowdsourced, anything that's doab a terminal is fair game. For example, cracking a password protected file or building a linux kernel.
- NL2Repo: Human annotators reverse engineered 104 open-source Python rep into a natural language requirements doc. The task for the AI is to recreate th repo given the doc

GDPval and non-coding agentic benchmarks

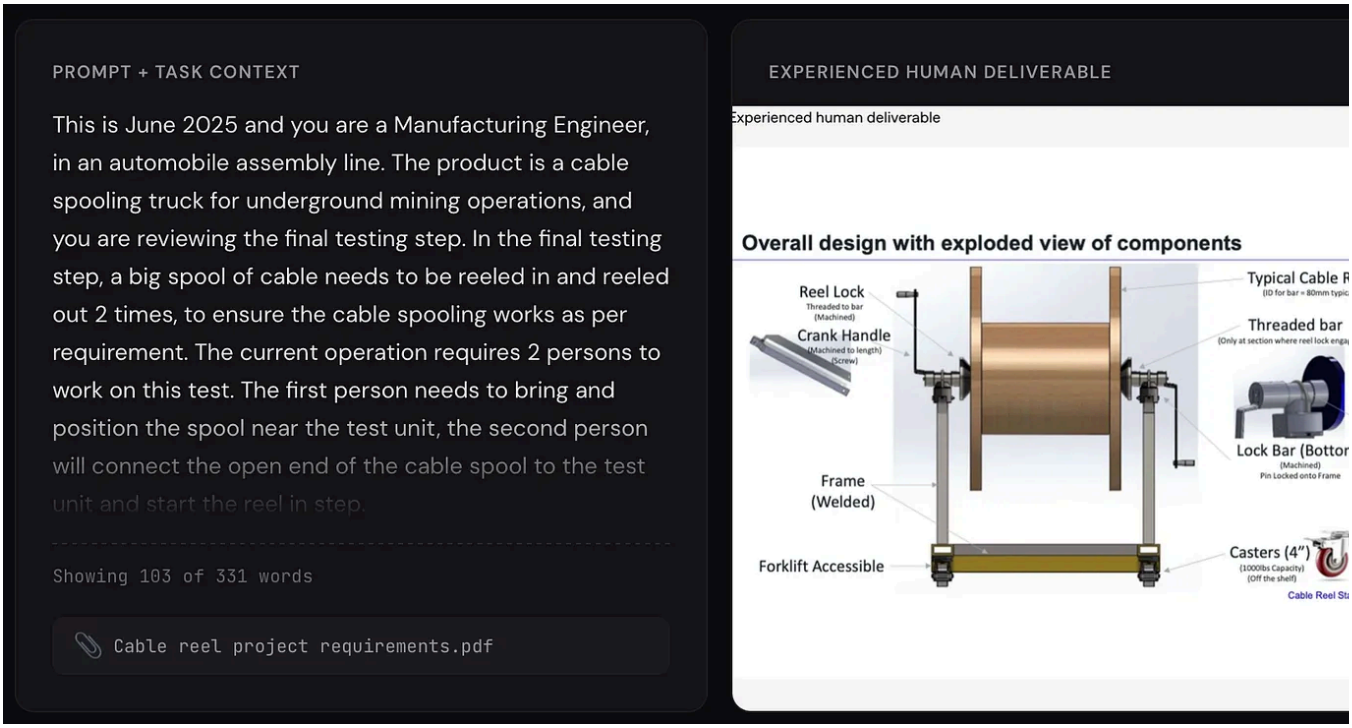
Agentic AI extends far beyond coding today, and so too do agentic benchmarks. T most famous example is GDPval by OpenAI. Released in September 2025, it aims measure AI's ability to complete real economically valuable tasks across 44 differe jobs, from financial analysts to nurse practitioners.

To create the tasks, OpenAI hired expert contractors from each job–e.g. an ex-Bof banker for finance tasks–and asked them to provide 3 things per task:

1. The problem statement, which can include reference files along with plain tex
2. An example solution to the problem, with deliverable formats spanning pdfs, spreadsheets, videos, etc.
3. A **rubric** that explains how to grade any given solution

The harness is another step up from coding benchmarks. Agents are given access apps like LibreOffice (Microsoft Office clone) and CAD software, along with the standard web search and code execution tools. Although GDPval isn't quite this advanced, newer agentic benchmarks also include fake calendars, emails, Slack messages, Google Drives, etc. that the AI needs to navigate in order to successfully complete the task.

Finally, to evaluate each task, OpenAI used additional expert contractors to compa the AI outputs to the human-provided solutions. They also created an AI grader th uses the rubric to rank solutions, but conceded it's still not as reliable as the huma experts. For this reason, they still use human experts for their official results, desp being way slower and more expensive.



Example GDPval task. Source: OpenAI

However, "LLM-as-a-judge" for evals is a popular technique used by other agentic benchmarks for tasks that aren't objectively verifiable. GDPval-aa, for example, is the public GDPval tasks but with an LLM judge.

In theory, rubrics allow you to measure important qualitative traits like style, but t have obvious limitations. For example, it's hard to guarantee quality when your ru are either written by contractors or AI generated. Using an LLM to evaluate quali also inherently suspect, especially when there's no human in the loop making the decision.

Another big limitation with GDPval is the clearly defined, unnaturally specified prompts. Real world tasks typically have an element of ambiguity that's completel missing from this benchmark. Human jobs also involve iteration based on feedbac whereas GDPval is strictly single-turn.

That being said, GDPval is certainly closer to actual knowledge work than someth like HLE. Other popular agentic benchmarks include:

- Apex Agents: Mercor benchmark that focuses exclusively on banking, consult and law. Tasks are created by their contractors. Agent is placed in a Google

Workspace environment complete with fake files, emails, etc. Uses LLM judg grading.

- Finance Agent: Tasks are created by human experts and involve analyzing rec SEC filings. Rubrics are generated by GPT-4o and then reviewed by humans. LLM judge for grading.
- BrowseComp: Tasks are hard to Google questions created by contractors. For example, "Between 1990 and 1994 inclusive, what teams played in a soccer ma with a Brazilian referee had four yellow cards, two for each team where three the total four were not issued during the first half, and four substitutions, one which was for an injury in the first 25 minutes of the match."
- OSWorld: Computer use benchmark that tests the AI's ability to use apps like LibreOffice, GIMP, and VLC. Tasks were manually created by 9 CS students v were listed as co-authors in exchange. Evals are custom scripts that check if t computer is in the correct state
- Tau-bench: Customer service benchmark that tests the AI's ability to do thing like cancel orders and modify flights. Environments and tasks were created by Sierra's researchers, but they used AI for things like fake data generation. Eva check the state of the application as well as the AI output for an exact string match.

Some sneaky benchmark reporting by OpenAI

Hopefully the previous sections have convinced you that benchmarks are often wi unrepresentative of the capability they claim to be measuring. However, they also definitely aren't totally useless, and a 10%+ improvement on SWE-bench verified a everyone else thought the benchmark was already saturated (which is what Mytho did) still means something.

Looking at the benchmarks companies choose NOT to report can also be telling. I example, OpenAI barely included any benchmarks in their GPT-5.4 a and didn't compare it to any Anthropic models. We think this is beca gotten brutally mogged by Opus 4.6—which came out a month earlie

our overall vibe of the model. Until yesterday, OpenAI's models were worse than Anthropic's for ~all agentic tasks.

	GPT-5.4	GPT-5.3-Codex	GPT-5.2
GDPval (wins or ties)	83.0%	70.9%	70.9%
SWE-Bench Pro (Public)	57.7%	56.8%	55.6%
OSWorld-Verified	75.0%	74.0%*	47.3%
Toolathlon	54.6%	51.9%	46.3%
BrowseComp	82.7%	77.3%	65.8%

Source: OpenAI

With GPT-5.5, they're finally back on the frontier, which is why Claude and Gemi were re-included in the benchmark table.

	GPT-5.5	GPT-5.4	GPT-5.5 Pro	GPT-5.4 Pro	Claude Opus 4.7	Gemini 3.1 P
Terminal-Bench 2.0	**82.7%**	75.1%	-	-	69.4%	68.5%
Expert-SWE (Internal)	**73.1%**	68.5%	-	-	-	-
GDPval (wins or ties)	**84.9%**	83.0%	82.3%	82.0%	80.3%	67.3%
OSWorld-Verified	**78.7%**	75.0%	-	-	78.0%	-
Toolathlon	**55.6%**	54.6%	-	-	-	48.8%
BrowseComp	84.4%	82.7%	**90.1%**	89.3%	79.3%	85.9%
FrontierMath Tier 1–3	51.7%	47.6%	**52.4%**	50.0%	43.8%	36.9%
FrontierMath Tier 4	35.4%	27.1%	**39.6%**	38.0%	22.9%	16.7%
CyberGym	**81.8%**	79.0%	-	-	73.1%	-

Source: OpenAI

However, there's still one benchmark that's suspiciously missing. Coding is the mo important model capability and OpenAI literally wrote a blog post in February arg for SWE-bench Pro to become the industry's new de facto benchmark. So why did use this random "Expert-SWE" benchmark instead?

Scrolling down all the way to the very bottom of the blog post reveals the answer:

Coding

Eval	GPT-5.5	GPT-5.4	GPT-5.5 Pro	GPT-5.4 Pro	Claude Opus 4.7	Gemini 3.1
SWE-Bench Pro (Public) *	58.6%	57.7%	-	-	64.3%	54.2%
Terminal-Bench 2.0	82.7%	75.1%	-	-	69.4%	68.5%
Expert-SWE (Internal)	73.1%	68.5%	-	-	-	-

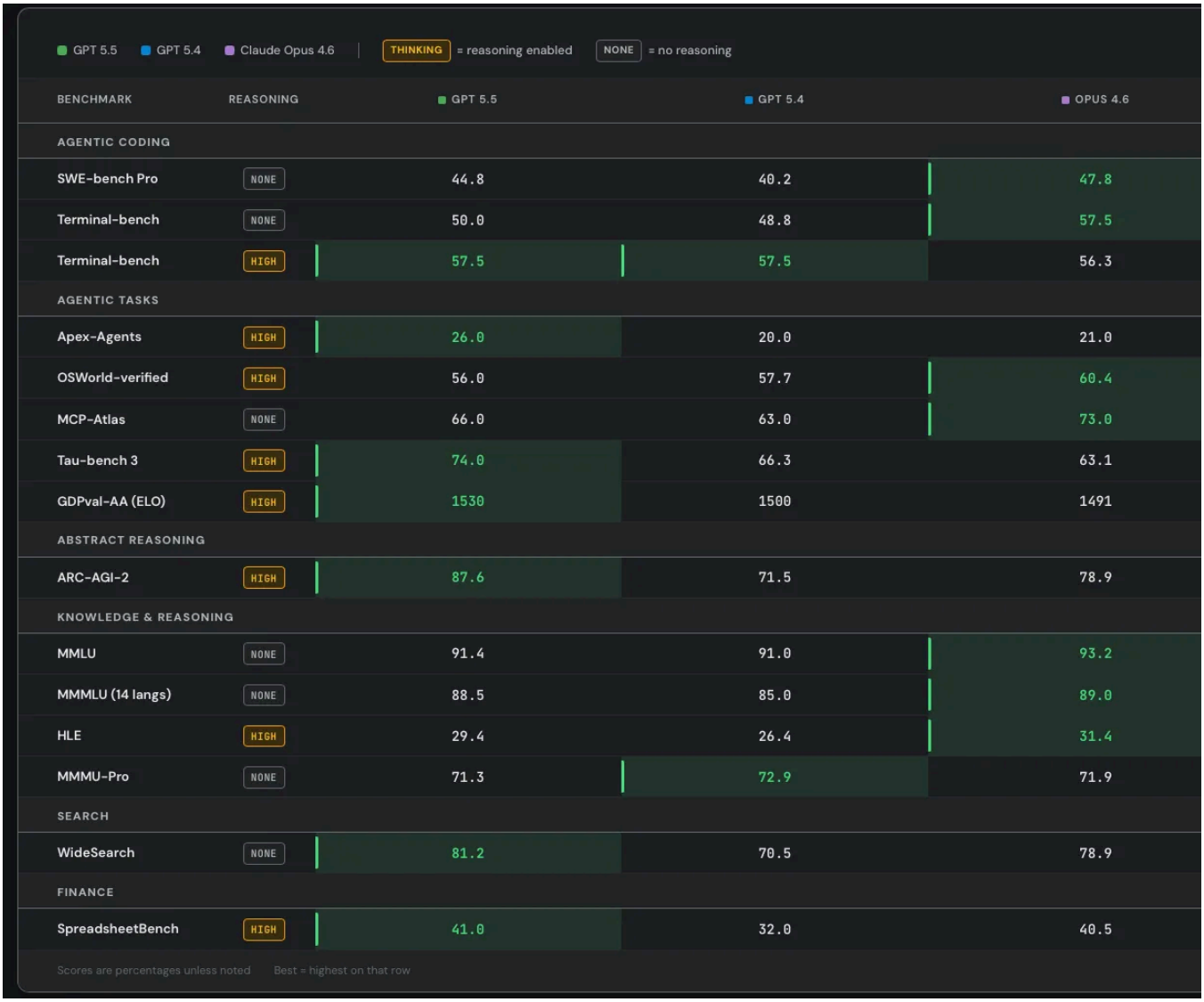
*Labs have noted evidence of memorization on this eval

Source: OpenAI

GPT-5.5 got mogged by Opus 4.7 (much less Mythos which scored 77.8%). This supports our qualitative impression of the three models. GPT-5.5 is better than Op 4.7 at some coding tasks but is not decisively better across the board. Mythos is presumably a true step up compared to both of them, but Anthropic hasn't given u access yet :(

Why you shouldn't use the same harness for an appl to-apples comparison

As part of our alpha-testing, we also ran a number of benchmarks on GPT-5.5 vs 5 Opus 4.6. Here are the results:



GPT 5.5 GPT 5.4 Claude Opus 4.6 | THINKING = reasoning enabled NONE = no reasoning

BENCHMARK	REASONING	GPT 5.5	GPT 5.4	OPUS 4.6
AGENTIC CODING				
SWE-bench Pro	NONE	44.8	40.2	47.8
Terminal-bench	NONE	50.0	48.8	57.5
Terminal-bench	HIGH	57.5	57.5	56.3
AGENTIC TASKS				
Apex-Agents	HIGH	26.0	20.0	21.0
OSWorld-verified	HIGH	56.0	57.7	60.4
MCP-Atlas	NONE	66.0	63.0	73.0
Tau-bench 3	HIGH	74.0	66.3	63.1
GDPval-AA (ELO)	HIGH	1530	1500	1491
ABSTRACT REASONING				
ARC-AGI-2	HIGH	87.6	71.5	78.9
KNOWLEDGE & REASONING				
MMLU	NONE	91.4	91.0	93.2
MMMLU (14 langs)	NONE	88.5	85.0	89.0
HLE	HIGH	29.4	26.4	31.4
MMMU-Pro	NONE	71.3	72.9	71.9
SEARCH				
WideSearch	NONE	81.2	70.5	78.9
FINANCE				
SpreadsheetBench	HIGH	41.0	32.0	40.5

Scores are percentages unless noted Best = highest on that row

Source: SemiAnalysis Tokenomics Team

Our numbers are generally lower than OpenAI's and Anthropic's for 2 reasons:

1. Both these labs use custom, closed-source, harnesses for their benchmark run order to increase performance
2. We only ran a subset of tasks for most benchmarks to save money. In some ca these subsets weren't representative. For example, for MCP atlas, we only considered 21/36 MCP servers and ignored tasks that required things like MongoDB, twelvedata, or alchemy.

You could argue that our benchmark numbers are better than OpenAI/Anthropic because we use the same harness for a more apple-to-apples comparison. But the harness is clearly part of the product at this point. What people actually care abou how good is Codex vs Claude Code, not GPT-5.5 vs Opus 4.7.

Returning back to the importance of token efficiency, it's worth emphasizing that **harness has a huge impact on the ultimate cost per task**. Prompt caching, input/output ratio, and tool use patterns are all largely determined by the harness. SemiAnalysis is currently collecting millions of dollars worth of agentic AI traces order to better understand how different harnesses (e.g. Claude Code vs Codex vs Cursor vs OpenCode) change cost per task. Preliminary analysis shows that Codex likely more token efficient than Claude Code, with an average input/output ratio o 80:1 vs 100:1. Yes, a higher input/output ratio means a lower price per Mtok, but C still ends up being cheaper because it consumes less input tokens overall. Those interested in the full results should subscribe to the Tokenomics model.

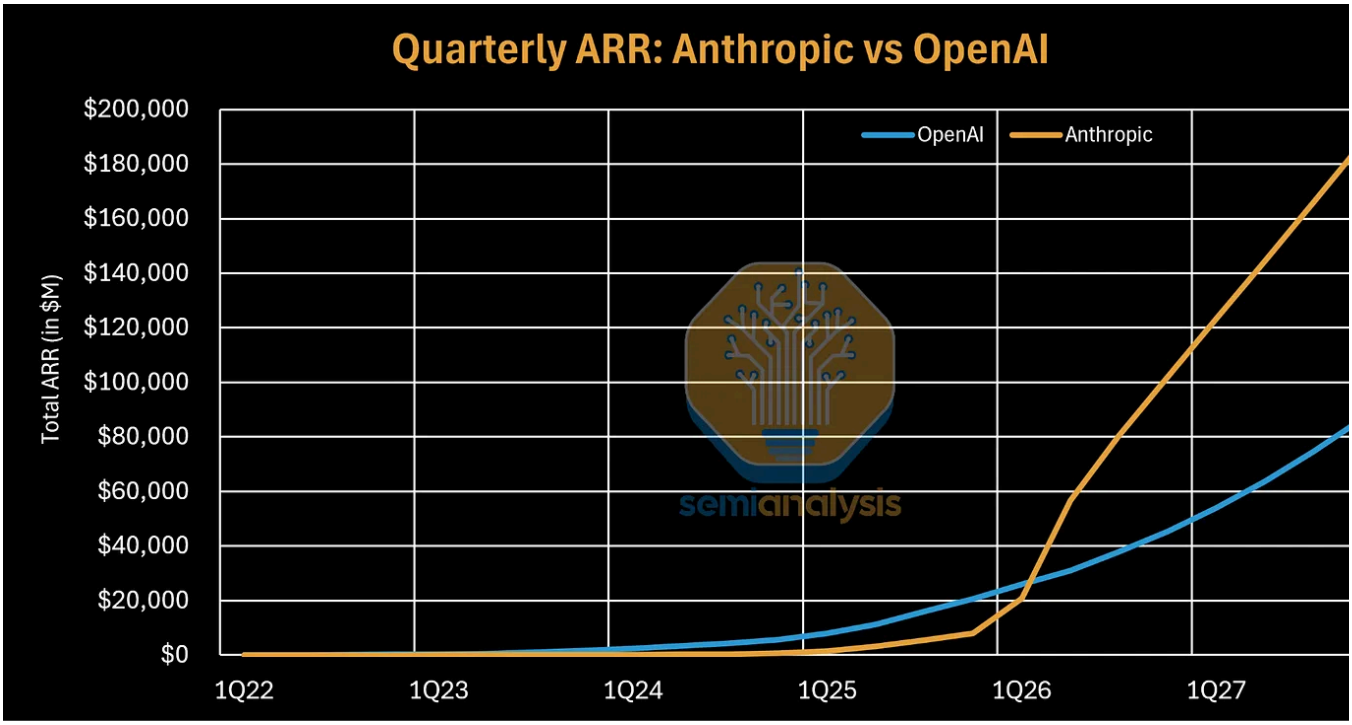
Who Wins in the Agentic Coding Wars?

So what does this all mean for the future of coding agents? Behind the paywall, we give our predictions for this soon to be multi-trillion dollar industry.

It's currently a two horse race between OpenAI and Anthropic. DeepSeek did put an incredible alpha drop in their newest v4 paper, but the gap between closed and open-source is starting to widen again. We think unmentioned in this race for *cod agents* is SpaceXAI, who after the recent Cursor pseudo-acquisition, has a shot at But it will take a genuine Elon miracle to displace Anthropic or OpenAI. Google h the resources to shock the world if they get their act together on RL, and Meta's M Spark debut puts them in the race but they're both still clearly behind.

To date, the only companies with any meaningful market share are Anthropic and OpenAI. Vibe coding startups like Windsurf/Cognition, Replit, Vercel V0, Lovable Kio, Base44, and others are all falling behind, even while growing revenue 3x or m in months. These companies often have negative gross margins, and their status a wrappers seems clearer than ever after the rise of Claude Code and Codex. The rea complete product is a harness and a model. Without one of the two, you're missing and the harder product is the model. The ARR of all of these companies added together is still in the low-to-mid single digit billions. Whereas we believe the majority of Anthropic's explosion from $9B to $40B (our current estimate) is attributable to agentic coding in Claude Code.

Things were looking bleak for OpenAI the past few months to put it lightly. Accounting discrepancies related to how the two companies recognize rev share f cloud providers aside, we believe Anthropic has truly passed them in ARR on a lik like basis. This outcome would've been inconceivable to basically everyone at the of the year, and we think most people still haven't fully internalized it. There's a n leader in the world's most important race, and their name is **Anthropic**.



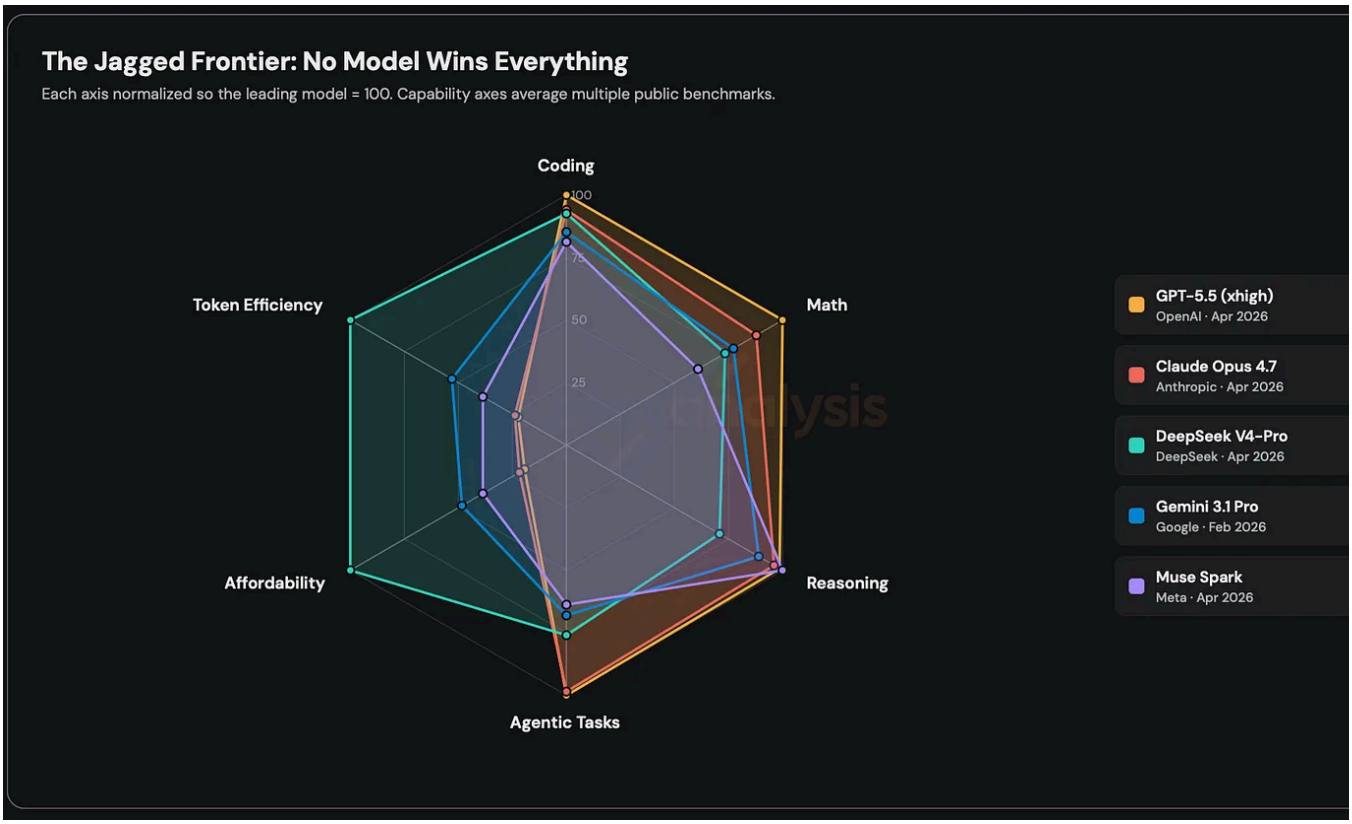
Source:SemiAnalysis Tokenomics Model

The lead is defensible because Anthropic's revenue is of higher quality. ChatGPT' free tier is generally known to be more generous and less restrictive than Claude's which is why OpenAI dominates on the consumer side. Anthropic's revenue mix i inverse with ~70% of revenue coming from API usage which scales with deployme rather than sign ups.

That being said, OpenAI's "code red moment" might be ending soon. OpenAI wil hungry to reclaim market share. Codex 5.5 is clearly in the same class as Opus 4.7, industry wide compute constraints will be OpenAI's saving grace to retain market share. Anthropic is aggressively buying up as much compute as possible, which is of the main reasons why H100 rental prices continue to soar. But there's only so m capacity the world can physically bring online this year.

We believe Anthropic is too AGI-pilled to ever dedicate more than 50% of their compute fleet to serving customers, and they're already starting to intentionally alienate large swathes of the market with price discrepancy. If we are to use Courn competition as a framework, Anthropic is becoming Evian water. Premium capaci like Opus 4.6 fast, lower rate limits during peak hours, testing removing Claude cc from the $20/month subscription, and banning third party harnesses like OpenCla are all tactics they've tried so far to shift XPU capacity to higher margin workload We expect them to be even more aggressive in the future.

The wave of customers that get priced out of using Claude or its uptime, is OpenA opportunity. That isn't to say GPT-5.5 is a bad model, In fact it outperforms Opus many dimensions on the jagged frontier.



Source:SemiAnalysis Tokenomics Model, Model release cards

Of course, there's the whole host of tasks where GPT-5.5 actually outperforms Op 4.7 as well, but we'll have to wait and see if that still holds post-Mythos.

1 * We display as FP8, but note that MoE expert parameters use FP4, while most other parameters use FP8 as per below.

Recommend SemiAnalysis to your readers

Bridging the gap between the world's most important industry, semiconductors, and business.

Recommend

63 Likes · 8 Restacks

Previous

Discussion about this post

Comments **Restacks**

Write a comment...

© 2026 Dylan Patel · Privacy · Terms · Collection notice
Substack is the home for great culture