AI DIGEST

NVIDIA took Claude Opus 5 from 30% to 100% on ARC-AGI-3 without touching the model. The only difference is the agent harness

AI digest, Sunday 23.08 - the day "which model" mattered less than "which harness around it": NVIDIA took Opus 5 from 30% to 100% on agent architecture alone, and Claude Code users caught a lab turning the same lever without saying so.

Sunday, 23 August 2026

IN THIS ISSUE

TOPIC 1

NVIDIA took Claude Opus 5 from 30% to 100% on ARC-AGI-3 without touching the model. The only difference is the agent harness

NVIDIA Developer blog (21.08), announcement from @NVIDIAAI - 70 points on HN, a second thread for the analysis.

Figures from the primary source, verbatim from the subheading: "The research project elevates **Claude Opus 5 from a 30% model baseline to 100%** as part of the complete AVO agent system, showing that **system design - not model capability alone** - can unlock frontier-level long-horizon performance".

The system is called **AVO** (Agentic Variation Operators). Results:

WHAT	FIGURE
ARC-AGI-3 public set	**100.00 RHAE**, all **183 levels** across **25 environments**
Efficiency	**12% fewer actions** in the environment than VISTA
GPU kernels (same architecture)	500+ directions, 40 kernel versions, **+10.5% over FlashAttention-4** on DGX B200

The source this topic arrived through needs a correction. It came in via a post from @ClementDelangue (CEO of Hugging Face, 17.2k views): "**NVIDIA built its own coding harness to optimize CUDA GPU kernels** and achieved a 100% score on ARC-AGI-3". That reads as if a kernel optimiser happened to pick up a benchmark. The article itself says something else: AVO is a **general-purpose agent**, "For ARC-AGI-3, we connected **the same general-purpose agent** to a different task interface. **The underlying agent remains the same**; only the environment-specific tools and evaluation change". Kernel optimisation and ARC are two demonstrations of one architecture. Neither is a side effect of the other. Delangue's post carries no link to the source at all, so the figures here come from the NVIDIA blog.

Why it matters. The article states outright what every agent harness rests on: "evaluating a model **is not the same as** evaluating an agent". The practical takeaway: when something fails, the first hypothesis should be the **harness** - what is in the context, which tools are available, how errors are caught, whether there is memory between steps. AVO has four mechanisms: persistent memory, supervision, tool-use and **recovery from failure**. The last one is missing from most homemade harnesses, and without it a failure in a background run passes in silence. [proven - vendor figures, no independent check from the ARC team on the public leaderboard yet, see "what I could not verify"]

TOPIC 2

Users say Anthropic shrank the effort scale in Claude Code without telling anyone. 175 points on HN, silence in the changelog. The

evidence is weaker than the headline

Thread by @argofowl, HN - 175 points, 27 comments.

The claim, verbatim: "anthropic enrols **fable 5** sessions on claude code **2.1.236+** into an experiment that shrinks the effort scale, older versions and **opus 5 are left alone**... if "high" feels like "low" for you, you're in the test group". And separately: "since 2.1.237 the model reads "high" effort as **10 out of 100**, the exact number "low" used to be - and the changelog doesn't say a word".

What was checked. The Claude Code changelog was pulled through the GitHub API (raw, not the rendered page), entries **2.1.235 - 2.1.241** read. On the effort scale, **not a word**: 2.1.237 is "Fixed prompt caching for sessions using an LLM gateway" plus the new Concise output style; 2.1.240 and **2.1.241 (shipped today at 03:52 Kyiv time)** are both "Bug fixes and reliability improvements". The part of the claim that says the changelog is silent is **confirmed**.

The measurement itself holds up poorly. The top HN comment lands squarely on the method: "the evidence, as best I can tell from the tweet, is that they **asked Claude what effort level it was set to**. But **how would the model even know that?**" The thread then splits: one side says effort is passed in the system prompt so the question is valid, the other says that is true of Qwen but not of most models. The measurement was made by **asking the model about itself**, which is exactly the class of evidence already labelled "indicator instead of data".

The soberest proposal in the thread comes from @toolshed_labs: "If enrolment is gated on the client version, then **pinning the previous client and rerunning the same prompts is the test**. Everyone comparing this week to last week **moved the client and the server at once**". That is the only way to tell a real change from a feeling.

Why it matters. Changing anything blind is premature. The claim concerns **Fable 5**. **Opus 5** is untouched by it. Practically: if a session seems to have "got dumber", Claude Code has `/effort`, and what to check is **behaviour on identical prompts**. Asking the model how it feels gets you nothing. [fuzzy - the experiment itself is unconfirmed; all that is confirmed is the changelog's silence]

TOPIC 3

The same plot from a third angle: identical weights on one GPU produce different tokens, and the backend is to blame

Level1Techs, a long read with experiments - 231 points on HN, thread.

The author takes the **official BF16 checkpoint of Qwen3.6-27B** on a single RTX PRO 6000 Blackwell, turns off everything that could confuse the picture (eager execution, no CUDA graphs, no prefix caching, no weight or KV-cache quantisation), and measures how far different backends diverge on which token comes next. The result: "For the first several thousand tokens, **every run of the model agreed**... Then **in later portions of the prompt,**

backends began disagreeing". The divergence shows up **on long context**, which is exactly where agent work lives.

The author's advice on measurement: "Do not crank temperature to zero and paste in 3 test prompts then call it good/bad. **Zero-shot tests are not a good analog of most agentic tasks.** You need **long-context tool-calling** and domain specific knowledge evaluations".

Why this sits next to item 2. Three separate stories in one day say the same thing: "**the same model" is an illusion**. At NVIDIA the harness made the difference (30% to 100%), in Claude Code people suspect a server-side experiment, here it is a different backend on the same hardware. The practical conclusion: "**feels worse" is not a diagnosis**. Either the measurement is made on identical prompts with long context and tool calls, or no conclusion is drawn at all. [proven - the author's own measurements with a described method and charts]

TOPIC 4

Levie: the bottleneck in AI diffusion is evals. And next to it a figure: $100M a year on inference with no offline evals at all

@levie (CEO of Box) - 14k views: "AI diffusion is **far more rate limited by having good evals** than most realize. The kind of evals that you see for every model release are incredibly helpful, but only tell you the shape of general AI progress and the relative capability level of models".

He quotes @BrendanFoody: "It is insane how many enterprises I meet that are spending **$100M / year on inference** and **don't have offline evals** to determine which model to use".

Why it matters. The same thought as in items 2-3, seen from the business side. The typical situation in a small automation: no evals at all, quality measured by whether anyone complained. That works with one user and nowhere beyond. The cheapest first step: fix a dozen typical requests with an expected result and run them whenever the model or the prompt changes. Without that, any "it got better or worse" stays a feeling. [proven as a claim about practice; the $100M figure is a practitioner's assertion with no sample disclosed]

TOPIC 5

Following yesterday: Ox Alpha, yesterday a "whose model, unknown", has now been tested by Mollick and turns out not to be frontier

Yesterday (the item on Inkling and misc) Ox Alpha went through marked "whose model this is has not been announced and has not been guessed". Today there is a measurement.

@emollick, 39k views: "I am **not as blown away** by the mystery model Ox Alpha as people appear to be. In my very early experiments, it seems fine, but **not at the frontier even among open weights**". The test is his standard WebGL "neo-Gothic city" shader, the same one he ran Kimi K3 through. Seven hours later he confirms: "This is a **consistent view**

across every test I am running. Nice model, but not frontier and **I am not sure why there has been so much buzz** as if it is".

Context for the noise: OpenCode is giving it away free for a week - 1M context, multimodal, zero data retention, "capacity for 100T tokens per day".

Why it matters. Loud free access is not quality. Time spent migrating to a stealth model is worth it after someone with a known method has measured it. Mollick is that case: he has run the same test for months, so his "not frontier" carries weight. [promising - one reviewer with one test, but a consistent method]

TOPIC 6

The new MCP roadmap: agentic primitives, one HTTP transport and agent identity in place of pasted API keys

Official MCP blog (22.08, by Lead Maintainers David Soria Parra and Den Delimarsky), HN - 183 points.

Five directions, three of which are actually on topic:

• **Agentic messaging primitives** - "Modern agentic workloads **no longer fit the standard request-and-response pattern**". That means server-initiated events (webhooks and channels), "so clients **aren't left polling** for results", plus maturing the Tasks extension (SEP-2663).

• **HTTP-native transport unification** - one transport for everything, including local servers speaking Streamable HTTP over stdio.

• **Agent identity** - the important one: "MCP authorization today is built around **a person approving access in a browser**... but more and more of the callers are **agents running as cloud workloads with their own identity**, acting on behalf of a user **who isn't present**". The solution is being built on DPoP, Workload Identity Federation and token exchange - "rather than **pasted API keys and long-lived tokens**".

Why it matters. That last point describes the state of any scheduled automation: the agent runs **with nobody at the browser** and rests on long-lived tokens and cookies in a browser profile. Every "the session expired" story is this. The standard is not ready, so for now this is only something to watch; but once DPoP and delegation land, that is the path scheduled runs should take instead of pasted keys.

The top comment in the HN thread is about **code mode** (Cloudflare) as an alternative: instead of loading every MCP tool into context, let the model write code that calls them. The counter-argument: "progressive discovery - kind of late to the party".

TOPIC 7

Munder Difflin: an open-source harness for running "an office of your own clones". #1 on GitHub Trending, 3,696 stars, 263 points on HN

Site, repo (MIT, 410 forks), HN thread.

The idea: a wrapper over **CLI agents you already have installed** - "works with your **existing subscriptions** (uses hourly limits)", supporting 12 providers (Claude Code, Codex, Gemini CLI, OpenCode, Cursor, Copilot...). Clones "capture workflows" and share memory: "Every clone you run **shares that memory**, so the next one you spin up **starts already knowing how you work**". Everything stays local: "Your code, your keys, your existing subscription - **nothing leaves your machine**". The only paid tier is Teams (a private 24/7 cloud and an E2E network between colleagues' clones).

Why it matters. The form factor is the one most homemade agent setups already have: a CLI agent on your own machine, on a subscription you already pay for, with memory shared between sessions. Munder Difflin adds a **parallel office of clones** on top. One thing here is worth noting: #1 of the day on GitHub Trending confirms that a local harness on your own subscription has become mainstream. The thread also carries a blunt "this project is cringe", so there is no consensus.

TOPIC 8

Mollick: Codex and Claude Code now fill in forms from your email unsupervised. And he writes out the condition that makes it work

@emollick, 21k views: "In terms of everyday usefulness and saving time, **Codex & Claude Code are very capable** of doing the thing where you ask them to "**fill out the forms that I got an email about**" and they do it well & **without further intervention** from you. Really nice for **low-risk** time-consuming stuff".

And in a separate post, the caveat that usually gets lost in the retellings: "To make this work you need to run the ChatGPT or Claude apps on your computer & **turn on browser control, which means you need to trust the models to do that**. Definitely **check the work carefully first** until you understand them".

Why it matters. For anyone who has already let an agent into a logged-in browser, the key word here is **low-risk**. The line runs along reversibility: reading, gathering, drafting, yes; pressing "pay" and sending mail under someone else's name, no. Mollick sets that condition himself, in the same thread.

TOPIC 9

Cursor opens Grok Bot to companies, the topic deliberately held back yesterday as "no announcement and no numbers"

@mntruell (CEO of Cursor) - **522k views**, the loudest post of the day in both lists: "We're **opening up Grok Bot access to a small set of enterprises this weekend**. Reply if you'd like for us to swing by your office in San Francisco and onboard your team tomorrow or Monday".

A clear bridge. Yesterday Grok Bot sat in the "deliberately skipped" section with this wording: "the 'channels for agents' form factor is interesting and fits yesterday's item on Slack Code, but this is **a discussion in three posts with no announcement and no numbers**. Keeping an eye on it". Today there is an announcement from the CEO himself, so the topic comes up. **There are still no specifics**: what the bot does, how many companies, the price - none of it. The volume comes from the poster's job title and the "we'll drop by your office tomorrow" format. [fuzzy]

TOPIC 10

Brockman on the pace: "a year ago 10 billion tokens a year got you a plaque, now that is a week"

@gdb, 72k views, quoting **@xeophon** (Florian Brand): "Not even a year ago, **10B tokens in a year** was a big achievement and got you a plaque. Now I do **>10B tokens a week**".

The scale explains the rest of the issue: when volume grows 50x in a year, labs get exactly the incentive discussed in item 2 (economise on compute without saying so), and companies get the problem Levie describes in item 4 ($100M on inference blind). This is an enthusiast's claim about his own consumption. There is no reported figure behind it. [fuzzy]

misc: @awilkinson built a "Bookshelf" on his site with Claude Code - his most-highlighted Kindle books plus his own newsletter notes, 23k views · Simon Willison: reviewing every line is not the only way to validate an agent "Eyeballing every line of code has never been the most effective way to validate a change" · @emollick against ELI5 by default - 52k views: "you aren't five! You don't need things **dumbed down**, you need things **explained differently**" - a better framing is "explain it building on what I already know" · @amasad: ""Pretty soon" turned out to be 3 months" 119k views · Replit in one week: 7 releases including skill import from GitHub · @patrickc shows Inkling launching for free 118k views - yesterday's item 7, now a one-line command · @avlok: at SpaceX 75% of shareholders take stock instead of cash against the usual 2% · @levelsio on German taxes - 236k views, "Germany does NOT take 50%. They take 49.3%", OECD source · @bentossell: "i never update chrome, i always update my agent apps" · hdiutil declared deprecated in macOS 27 173 points - thread, affects any script that touches .dmg

---