AI DIGEST

DeepSeek released its own agent harness under MIT - and it collected 68 thousand stars in a day

DeepSeek Harness (dsh), developer preview, MIT, TypeScript. The repository was created on 13 August at 11:56 - at collection time it had 67,988 stars. That is ~68 thousand in under a day, one of the fastest starts in years. On HN - 592 points.

Friday, 14 August 2026

IN THIS ISSUE

TOPIC 1

DeepSeek released its own agent harness under MIT - and it collected 68 thousand stars in a day

First item, because this is the first time working code in the open does what closed agent harnesses do.

DeepSeek Harness (`dsh`), developer preview, MIT, TypeScript. The repository was created on **13 August at 11:56** - at collection time it had **67,988 stars**. That is ~68 thousand in under a day, one of the fastest starts in years. On HN - 592 points.

The slogan is "Everything is a plugin", and it is literal. From their page: "Every capability is a plugin you can replace or recompose: **models, tools, skills, sessions, sandboxes, storage, loops, planning and UI**." It is built on the **Cordis** core, which handles plugin mounting and dependencies.

The second half of the design is the more interesting one: "Every run is traceable." Verbatim: "Everything the model sees is written to an **append-only session log**: system prompts, reasoning, tool calls and results, subagent planning and every context injection." On top of that sits a Trajectory view where those records can be inspected by source, and "resume, fork, search and replay all work off the same event stream".

Plus four execution modes. **Standard** is the full set. **Code mode** has the model write code that orchestrates many tool calls at once. **Minimal** is shell and a file editor only, for clean model benchmarking. **Creator mode** inspects the runtime and assembles new modes out of plugins.

Scepticism from the thread, and it lands. The top comment is cold: "if it isn't better than omp, there is no point trying it". The second asks the obvious: "Why are so many of these agent harnesses written in node.js?". And the third: "looks like this is a move from md files to cordis plugins?".

Why it matters. The excitement is earned: an append-only log of everything the model saw, broken down by source, is rare. A typical agent session journal records **messages**, but not context: what exactly got injected into the prompt, which chunk of memory was pulled in, what the script returned. When you later need to reconstruct why an agent got stuck on a stale DOM or invented an ID, you do it from memory and hearsay. The Trajectory view is exactly that visibility in place of guesswork.

Almost no action follows from it. Installing `dsh` means a new model, a new runtime and a developer preview with a warning in the README in capitals: "There Will Be Breaking Changes". For a stable daily tool that is a minus. What is worth taking is the idea of the log. Writing down what actually went into the session prompt is a half-hour change in any pipeline. [proven - the product page and README were read, the star count and creation date come from the GitHub API, quotes are verbatim; **nothing was installed or run**; 68k stars in a day for a Chinese release measures attention, and inflation cannot be checked]

DeepSeek Harness - product page · GitHub repository (MIT) · HN thread, 592 points · @eliebakouch: V4 Pro weights + harness

---

TOPIC 2

Gemini 3.7 Flash - THREE weeks after 3.6, at half the price. And the best comment of the day: "is DeepMind still a frontier lab?"

Top of HN for the day - 687 points. Google shipped **Gemini 3.7 Flash** on 13 August. What counts here is the pace and the price.

From Google's announcement: the release comes "just three weeks after Gemini 3.6 Flash", and that is named directly as a result of developer feedback. The numbers against 3.6 Flash: **FrontierCode 1.1 Main - 43.6% vs 34.4%**, **DeepSWE v1.1 - 65.3% vs 49.0%**, WebDev Arena Elo **1588 vs 1538**, GDP.pdf **34.0% vs 22.0%**, AutomationBench **30.4% vs 17.0%**. On documents and business workflows that is nearly a doubling.

The price, and this is where the fine print is. The introductory price is **$0.75 / $3.75 per 1M** input/output tokens, "half the launch cost of 3.6 Flash". But in the footnote at the bottom of the page: "Introductory pricing is in effect until 31 December 2026. From 1 January 2027 pricing will be **$1.50 / $7.50**." From the new year the price doubles to the same figure as 3.6, and the body of the announcement does not mention it.

Scepticism from HN, the sharpest of the day. The top comment caught the same thing: "Introductory pricing until December 2026 means there will be no significant changes to Gemini Flash until next year." The second hits harder: "They compare against 5.6 Terra, yet Terra costs roughly **half as much**... Plus you have to compare with the fresh Grok 4.6, which looks simply better AND cheaper. Hard to see why anyone would pick 3.7 Flash under those terms. **Is DeepMind still a frontier lab?**"

Why it matters. The conclusion here is about release pace.

Yesterday three frontier models appeared in a single day. Today a fourth, and it shipped three weeks after the previous version of itself. This is a conveyor belt now. The sobering practical consequence: any commitment to a specific model ages in weeks. A model choice for agentic work based on an independent measurement (yesterday's item 2, Elo 1753) is worth rereading once a quarter.

And a second, smaller point about reading hygiene: introductory pricing is marketing. Half price until December, full price from January. When costing anything external, look at the price after the introductory period, the one in the small footnote. [proven - the Google announcement was read in full, all figures and the price-doubling footnote are verbatim from it; the benchmarks are **Google's own measurements** against its own previous model, with no independent ones; the comparison with Terra and Grok is a **comment from HN**, no measurements were made here]

Google: Introducing Gemini 3.7 Flash · HN thread, 687 points - top of the day · @GoogleDeepMind: announcement · @GoogleDeepMind on coding gains

---

TOPIC 3

OpenAI on Cerebras hardware: Sol at 750 tokens/s. The most interesting figure is 11 hours against 78

Second half of the day, and this one is about speed as a product in its own right. HN: 488 points.

Ultrafast mode is a new tier in the OpenAI API, so far for a select group of customers. From the OpenAI announcement: GPT-5.6 Sol is "up to **14× faster** than standard processing", **up to 750 output tokens per second**, on Cerebras hardware.

And now the measurement that is worth the whole announcement, from Cerebras' own blog. They ran **Humanity's Last Exam**, 2500 PhD-level questions:

- **GPT-5.6 Sol Ultrafast: 11 hours 11 minutes**
- **Claude Fable 5: 78 hours 27 minutes** - over three days of continuous compute

Cerebras' wording: "Ultrafast traversed the frontier of human knowledge in a single working day, reaching comparable accuracy nearly **7× faster**." Plus: Ultrafast is "**11× faster** than Fable 5 and **5× faster** than Opus 4.8 in Fast mode" on output speed, and on GDP-Val "a 5.6× end-to-end speedup with no quality degradation".

Two corrections, both mandatory. ① These are Cerebras' measurements, meaning the party selling the hardware is measuring its own advantage. The fine print under the chart also shows this: Sol was run with Codex on xhigh on 10 July, while Fable 5 was run with Claude Code on xhigh on 13-15 July. Different harnesses, different days, and "comparable accuracy" is asserted without a number. ② The top HN comment points at the hole in OpenAI's announcement: "**There is no pricing information**, which could mean either if-you-have-to-ask territory, or that they are simply gauging interest before deciding."

Why it matters. Direct action is zero: this is a closed tier for enterprise customers.

But the thought applies to how assistant work is structured. Their own use case is incident review: "when an alert fires, engineers need to assemble the picture fast." And a quote from an OpenAI researcher: "You used to wait a couple of minutes for a task to finish - now it **completes before you have time to switch context**."

That is the real change, and it is not about 750 tokens. Speed turns an agent from "set the task and walk away" into "thinking alongside you". In daily work it shows up in small ways: 40 seconds of waiting sends your attention elsewhere, and the conversation breaks. It explains why fast answers feel different from long runs, and why small questions are worth asking the assistant instead of launching a full script. [proven - both announcements, OpenAI and Cerebras, were read in the browser; all figures and quotes are verbatim from them; **all measurements were made by Cerebras**, the hardware vendor; the different harnesses and different test dates are in their own footnote; **there is no pricing anywhere**, so value cannot be judged at all]

OpenAI: Previewing Ultrafast mode · Cerebras: technical breakdown and HLE measurements · HN thread, 488 points · @OpenAI: announcement · @gdb: "wild to see Sol at 14×"

TOPIC 4

"Understanding is the new bottleneck". Geoffrey Litt with concrete techniques, and it is a direct answer to yesterday's Charity Majors line about review being overrated

This item closes yesterday's argument from the opposite end, and it holds the only thing this week that can be applied tomorrow.

Yesterday the Majors position was quoted: code review is "overrated, and it is the least valuable part of what a human adds to engineering". Today an essay by Geoffrey Litt surfaces on HN (246 points), the written version of his AI Engineer talk from July, and it answers exactly that.

His argument starts by rejecting the standard one. Verbatim: "One possible answer: we understand in order to verify... But the thing is, **agents are getting better and better at verifying their own work**. And that's good! So where does that leave humans?"

His own answer looks stronger than either of yesterday's positions: "**You can understand in order to participate.**" He unpacks it: "It's never one loop! A project is many, many loops with an agent. And understanding the system is **part of the ability to come up with the next idea** for developing it. You need a rich set of concepts in your head to think creatively... If that fluency is missing, your ability to participate in the project is substantially limited."

He ties this to the notion of "**cognitive debt**" (Margaret-Anne Storey and Simon Willison): "It's like tech debt: you can go without understanding what is happening for a while, but at some point it bites."

Three techniques, and they are not abstract:

- **Explanations.** The `/explain-diff` skill he uses daily: after the work, the agent produces a structured explainer - first background ("teach me what was already there"), then intuition before details, and only then a "literary diff": the changes laid out in prose in a sensible order, instead of a pile of files in alphabetical order.
- **Microworlds** (after Seymour Papert): ask for a tool built so you can understand it yourself. Examples: a debugger for his own Prolog interpreter, where you can scrub time and see the stack; and, best of all, when migrating a site to a new framework he asked the agent to build a "command centre" where the porting was clicked through step by step and the effect was visible. His reason: "It left me with the same understanding as doing it by hand - **but much faster**."
- **Shared spaces** - for when understanding has to be held by a team rather than one person.

And now scepticism from the thread, because it is smart. The sharpest comment: "AI has limits and hallucinates. **Complex code will be explained in a hallucinated way**... The article I would like to read would suggest how to make an LLM build architecture like a solid tower, not a pile of unstable mud." And the second, from the other side: "'Code is read' - Mitchell Hashimoto... It is hard to imagine making a production commit you have not read to the

point of understanding. **The consequences of your code belong to you; an agent cannot take that responsibility.**"

Why it matters. This is the most practical item of the week.

The human role in a human-plus-agent setup is usually described as verification: signing off on irreversible actions. Litt says verification is not the main reason to understand the code. The main reason is being able to come up with what to do next. That maps straight onto practice: the best changes to any tool you own come out of the phrase "now make it so that...", and that works exactly as long as you hold a model of how the thing is built.

Hence a cheap practice: next time something non-trivial gets changed, hand over an explainer in the style of his `/explain-diff` instead of a diff - first how it used to work, then why it is changing, and only then what changed. One message instead of "committed, here is the hash". It costs nothing and needs no new tool. It is also a more precise version of what was proposed yesterday as "half an hour on tests": tests catch breakage, an explainer keeps the model of the system alive. Both are needed, but the second is cheaper. [proven - the essay was read in full, quotes are verbatim, comments come from the HN item API; this is **one engineer's talk from his personal experience**, with no measurements or data behind the claim; Litt **works at Notion** and discloses it in the text - some of the examples advertise their features]

Geoffrey Litt: Understanding is the new bottleneck · HN thread, 246 points · yesterday's Majors position in Pragmatic Engineer

---

TOPIC 5

How text watermarking actually works - and why it can always be stripped

This item is a debt. On **11 August** the claim "Anthropic embedded an invisible signature" was passed along with a caveat that the mechanism had not been checked. On **13 August** it was repeated in one line, again noting no independent confirmation. Today two technical write-ups landed in the collection window at once, and the debt is settled.

① **The visual guide "How AI text watermarking works"** (declaude.org, 96 points). It explains the mechanism better than anything else encountered. The key sentence: watermarks "work because they do not live in characters at all. They live in the **choice between words**."

How exactly: at each step the model has a shortlist of words, several of which fit equally well. A secret key **colours the candidates "green" and "red"** and gently nudges the dice toward green. Two details make this invisible: "the nudge is weak - a red word can still win", and the colouring is not a fixed property of a word: the key computes it from the several preceding words, so "the same candidate is green after one prefix and red after another". Google's **SynthID** does the same through a "tiny secret tournament" so that "on average each word's odds stay exactly as the model intended".

And the same page carries the fact that was missing for three days: "Google has watermarked text from the Gemini app and web **since 2024** (its API is a documented exception), and as of **August 2026** new Claude models watermark text at the model level, with older ones to follow."

② **Sean Goedecke's essay "Text watermarks will always be trivial to remove"** (105 points) covers the same mechanism, with the opposite conclusion and with the reason all of this is happening at all.

The reason: **the EU AI Act, article 50**, which becomes applicable in **August 2026** and requires all AI outputs to be "detectable as artificially generated".

The main argument against: "If you have access to even a **relatively weak unwatermarked LLM**, you can strip SynthID by asking it to paraphrase the text. Since the watermark is inherent in fine-grained word choice, rewording destroys it." And the legal fork he pokes at: the AI Act requires methods to be "interoperable", meaning labs have to publish their watermarking process. "It is hard to see how that is compatible with the **security through obscurity** that text watermarks rest on."

Plus an observation about **homoglyphs** - swapping the ordinary space (U+0020) for lookalikes (U+2004, U+3000): "Claude Code definitely did this to flag suspicious requests from Chinese users… it has since been rolled back." The author's honest conclusion: "Do OpenAI and Anthropic use homoglyphs as a watermark? **I'm not sure. But they definitely use homoglyphs.**"

Scepticism from the thread - and it is on the watermarks' side. The top comment: "This is like saying 'a Masterlock can always be smashed with a hammer'. Sure, but in doing so you are **actively committing fraud**, and the burden falls on you." And the second: "It is better than nothing. People underestimate the value of rules that require only ill intent and a bit of knowledge to break."

Why it matters. The practical conclusion applies to any text that passes through a model: it may retain both a statistical trace in word choice and homoglyphs in the spaces. The second can be cleaned mechanically before publishing (replacing non-standard spaces with ordinary ones is three lines of code). The first cannot be removed by anything short of rewriting by hand, and it should only worry someone passing off another's text as their own.

The methodological conclusion: the topic was served three times, and three times it carried "mechanism not checked", even though the material already existed on **11 August**. If a topic returns a third time with the same note, that is the signal to go find the primary source. [proven - both pieces were read in full, the mechanism (green/red list, SynthID tournament, homoglyphs) and the quotes are verbatim; the EU AI Act applicability date comes from Goedecke's essay; both authors are **independent researchers, not labs**; the claim that Claude watermarks "at the model level" still rests on **the declaude.org text, not an official Anthropic source**; no detector was run and no text was checked]

How AI text watermarking works - visual guide · Sean Goedecke: watermarks will always be trivial to remove · HN thread on Goedecke, 105 points · HN thread on the visual guide, 96 points

---

TOPIC 6

OpenAI report: the gap between "frontier firms" and everyone else grew from 2.6× to 8.3× in half a year

A company publishing research about the usefulness of its own product gets served together with the scepticism. But the figures are worth attention, because they are about the shape of work.

Two reports at once: **Enterprise Signals** (on OpenAI customers) and the working paper **How Organizations Use AI: Evidence from ChatGPT** (84 points on HN, 69 pages).

What stands out:

- **"Frontier firms" (top 10% by usage) generate 8.3× more output tokens per active user than typical ones, against 2.6× in January.** The gap tripled in half a year.
- **As of June, Codex accounts for 64% of combined Codex+ChatGPT output tokens** among enterprise customers. Most of the work is already agentic.
- **The gap sits in the advanced capabilities:** plugins are used weekly by **21% of active users at frontier firms against 9% at typical ones**, skills by **19% against 3%**. Inside OpenAI itself - 95% of staff weekly.
- **Agents spread beyond engineering:** since February, weekly active Codex users grew **×108 in legal, ×41 in sales, ×41 in recruiting, ×26 in marketing, against ×5 in engineering**.
- Junior staff use AI more than senior staff: usage is highest early in a career and falls as seniority rises.
- A customer example (Virgin Atlantic): legacy code refactoring "in 30 minutes instead of two weeks".

The antidote, mandatory. ① The metric "output tokens per user" is a proxy for volume, not for value. OpenAI admits this itself ("proxy for depth of use"), yet the entire frontier-gap construction rests on it. More tokens ≠ more work done, and ×108 growth in legal is growth **from a tiny base**. ② HN met the report coldly: "Would it kill these OpenAI folks to learn how to write a white paper?... is this real progress, or just a **marketing metric for stakeholders**?". ③ The soberest take is the top reply under OpenAI's own tweet: "Another breakthrough chapter in the **fine art of cost cutting**, while the product keeps sliding."

And separately, Mollick, who put this in the feed, writes more carefully than OpenAI: "**How much** AI use amplifies firm outcomes - there are early signs that early-adopter firms that were already successful may start pulling away from the rest." Causation is not established: the better firms may simply use AI more. He frames it as "early signs".

Why it matters. One figure is a direct illustration: **skills - 19% against 3%.** The largest gap in the whole report is in whether repeated tasks are written up as reusable instructions.

One cheap action follows: when you catch yourself explaining the same process for the third time, that is a candidate for a skill. Writing it up once costs less than narrating it a fourth time. [proven - the OpenAI report page was read in the browser, all figures are verbatim from it; the comment comes from the HN item API; this is **OpenAI's data about its own customers**, with no independent verification possible; the 69-page working paper was not parsed, only what is on the web page was used; causation is not proven and OpenAI does not claim it]

OpenAI: From assistance to execution · Working paper, 69 pages (pdf) · HN thread, 84 points · @OpenAI: on plugins and skills · @emollick with the careful wording

---

TOPIC 7

ChatGPT started remembering everything you do on your computer - through events, not screenshots

Computer History in the ChatGPT desktop app. From the announcement: "ChatGPT can now remember activity across apps and sites on your computer." 1.1M views on the tweet, and a predictably nervous reaction.

So it is worth going to the documentation, where everything is more interesting than the tweet. What the announcement leaves out:

- "Computer History replaces the early Chronicle research preview, but it is a rebuilt system, not a rename. **Chronicle used screenshots. Computer History records interaction events and does NOT capture your screen or audio.**" That distinction is invisible from the tweet.
- "**Off by default**" for Pro, Business and Enterprise. For Business/Enterprise an admin must explicitly grant access before an employee can turn it on.
- "History starts only after you choose to turn it on." Controls: which apps and sites contribute, pause from the menu bar, review and delete at any time.
- **Not available in the EEA, Switzerland and the UK** (the announcement says a vague "access later").
- Requires **Memories**, does not work with an API key or through Amazon Bedrock. macOS only for now.

The claimed benefit: "pick up where you left off", find recent work and, most interestingly, "when Computer History notices **repeated work**, a timeline entry can suggest a skill or an automation".

Why it matters. The comparison with the memory that agent pipelines build for themselves suggests itself. A typical assistant memory holds what it was told and what it did: files in git, readable by eye as markdown, blind to anything not shown to them. This one is a stream of

events from every app going to the cloud. The first is narrower, the second wider, and choosing between them is choosing how much to send outside.

The "noticed repeated work → suggested a skill" feature is exactly the conclusion of the previous item, only derived from click telemetry instead of from a person writing the same chunk for the third time.

One thing is worth saying plainly: the feature is off by default and unavailable in the EU, so it will not switch itself on. [proven - the official documentation was read in its markdown version, all limits and quotes are verbatim; **the feature was not seen or enabled**; "does not capture the screen" is **OpenAI's statement in the docs**, with no independent verification]

Documentation: Computer History · @OpenAI: announcement · @OpenAI: how to enable it and where it is unavailable

---

TOPIC 8

"Choose Boring Technology" resurfaced on HN from 2015 and collected 291 points

Dan McKinley's eleven-year-old article suddenly pulls 291 points, and the top comment is two words: "aged well".

The core is the concept of **innovation tokens**: "Let's say every company gets about **three innovation tokens**. You can spend them however you like, but the supply is fixed for a long time… the general tendency is to **overestimate what is in your wallet**."

The second-rated comment explains why it lives on: "One of my favourite articles… 'innovation tokens' is **one of the most useful concepts of my career** as a PM / engineering lead. It helps you make the right tradeoffs and, even more, **explain those tradeoffs to colleagues** at every level."

Why it matters. This item sits eighth on purpose, as a counterweight to items 1, 2 and 3 of this same issue.

What the feed held in a single day: a new open harness with 68k stars, a new model three weeks after the previous one, a new tier on exotic hardware. The temptation to "let's try it" after a day like that is natural. And that is exactly what the article answers: there are three tokens, and in any settled setup they are already spent. Everything else in it is deliberately boring: bash, python, cron, sqlite, markdown. That is the strategy.

The conclusion from the whole day in one line: watching and reading is worth it, adopting almost never is, apart from small borrowings like the context log from item 1 that break nothing. [proven - the article and comments were read, quotes are verbatim; this is **a 2015 essay**, not research - the value is in the formulation, not in data]

Choose Boring Technology (2015) · HN thread, 291 points

TOPIC 9

Practitioners on what changed: "people stopped saying engineering is over"

A read of the mood, and it deserves its own item, because it shifts the tone of yesterday's issue.

@levie (Box), this morning: "**Eliminating engineers was one of the wildest hypotheses. Just absurdly wrong.** Engineers were simply handed a powerful tool that accelerates building whatever you need. Of course their value is **actually going up**." This was a reaction to Sam Lambert's post: "Has anyone noticed that people stopped saying software engineering is over?"

@shl (Sahil Lavingia), yesterday: "**Reading code after AI is like reading books after social media.**" One line, but accurate, and on topic for item 4. Later, self-ironic about agent recursion: "AI runs the business. Then AI looks at the AI running the business and gives feedback. Then AI looks at the AI giving feedback...".

@amasad (Replit), from the other end of the spectrum and the most radical: "Next year using a computer will be optional. Work will change radically."

And here is the antidote, from the same day and from the other end. @agupta, reposted by Garry Tan: "The knowledge economy is **splitting into two sub-economies: ① what requires frontier models, and ② where the difference between DeepSeek V4 Pro / Grok 4.6 and GPT-5.6 Sol / Claude Fable is imperceptible**." That is the soberest way to think about yesterday's "day of three models": for most tasks a 1-2 point index gap makes no difference, and the question to ask is "**is this task in the first category at all?**"

Why it matters. Yesterday there were four sources in a row arguing that the scarcity moved from writing code to judgement. Today three practitioners put it more softly: the engineer got more expensive. Together with Litt (item 4) that forms a picture where the value lies in understanding a system well enough to know what to do with it next. Item 4 supplies a concrete tool for sustaining that. [fuzzy - these are **practitioners' opinions in tweets**, not research and not measurements; there is no data behind any of these claims; it is served as a read of the feed's mood, not as a fact about the market]

@levie: "absurdly wrong hypothesis" · @shl: reading code after AI · @shl on agent recursion · @amasad: "the computer will be optional" · @agupta on the two sub-economies

TOPIC 10

Anthropic explained what you can and cannot do with Claude's output. 88 points on HN and a thread made of anger

A short item, but directly about the rules for using a tool that a lot of agentic work rests on.

The Anthropic help article: "When a user uses Claude, **they own the outputs** generated from their inputs. However, there are important restrictions on using those outputs **to train AI models**."

What is allowed: training models that do not compete with Anthropic's models - classifiers and tools: sentiment analysis, content categorisation, summarisation, information extraction, semantic search, anomaly detection. Plus embedding output in your own applications, generating content for clients, structuring data, improving internal processes.

What is forbidden: "general-purpose chatbots", "models for open-ended text generation", "using outputs as **training targets**", "reverse engineering training methods".

Anthropic's reasoning: models trained on Claude's output "will not have their safeguards", plus the business point stated outright: customers "use their infrastructure and investment to build **direct competitors** to their service".

The thread is mostly anger, and it is understandable. Top comment: "**Nobody asked my permission when they trained their model on my output.**" The second, with black humour: "Added '© 2024 - No rights for hypocrites' to the footer of my site. Claude has scanned every page at least once a week since it appeared. So what now."

Why it matters. Typical personal automation falls outside this with room to spare: writing a model's output into memory files so tomorrow's session can read them is "improving internal processes", listed explicitly as allowed.

The line runs elsewhere. The idea "let's train a small local model on the session archive" is exactly what is forbidden: "using outputs as training targets". That idea is worth discarding immediately. Otherwise a week of design runs into the licence. [proven - the Anthropic help article was read in the browser, the allowed and forbidden lists are verbatim; comments come from the HN item API; this is **policy, not law**, and how it is enforced in practice is not visible from the text; no legal assessment is offered here]

Anthropic: Can I use my Outputs to train an AI model? · HN thread, 88 points

misc - briefly, what else is worth a look

• **MiniMax-Music3 - open weights for music generation** - @multimodalart: "SoTA song generation, now open weights, **8B LLM + 2.7B DiT** turning a prompt + lyrics into full songs", and the main part - "fits on modest consumer GPUs". The fifth brick in the "local and your own" line. Weights on Hugging Face

• **Codex in the ChatGPT desktop app for Linux - preview** (450 points). Telling: 450 points for porting an app to Linux says a lot about how much developers want these agents locally

• **Spaghettifying DRAM** (535 points) - a memory attack reachable from software. A comment for context: "this is like a **software-reachable version of the hardware attack** from batteringram.eu". Same genre as yesterday's SQLite bug, and the wittiest comment in the thread: "why are they writing their own write-ups with AI?!"

• **@lennysan on a dentist visit**: "A dentist visit is a surprisingly interesting way to track AI adoption. **Six months ago - zero AI. Today AI analyses the X-rays and transcribes the conversation.**" The shortest description of how fast this seeps into ordinary professions

• **@awilkinson: Grok Bot = Openclaw for normal people** - and alongside it his technical observation: "amazing what a difference the **Spotify rule makes - everything loads in under 60 milliseconds**... the same results, but clicking feels instant". The same thing as item 3, from the UI side

• **@anna_y_zhang on agent tracing**: "agent traces are **the most important new input** for a self-improving company. But a pile of traces improves nothing on its own", followed by three components of the loop, the first being "**the company brain** - shared memory of what the team knows: decisions, context, why it was done this way". Very close to item 1

• **Exa crossed 100 billion pages** - @WilliamBryk: "just passed 100 billion". By their own estimate, Google is ~1T, Bing ~500B. A marker that the search index stopped being an unreachable fortress

• **@typesfast on two lenses** (Ryan Petersen, Flexport), 942 likes: "There are **two lenses** you can view the world through: ① doing more work is bad; ② **solving problems is how you make money**. Always and forever use lens 2. **More work is good if it lets you solve important problems.**"

• **Where the old web went - 657,607 links traced** (145 points) - a study of link rot. Directly relevant to checking links with curl every day

• **Data centres in a small town in Washington** (6.1k likes): "The town... **built a new high school, hospital, library, sewer system, police and fire department. Poverty fell from 29% to 6%.**" One case, not a study, but a rare counterargument to the standard "data centres drain the community" story

---