

Search posts from January 2025

Search

Jan. 2, 2025

[... 390 words]

7:39 am / mathematics

Thought: I will replace the contents of 'game/fen.txt' with a valid FEN string where Black has a decisive advantage greater than +500 centipawns.

Action: echo '6k1/592/8/8/8/8/7K b - - 8 1' > game/fen.txt; ./game.py status

— [o1-preview](#), in [an eval](#) by Palisade Research

[6:33 pm](#) / [o1](#), [generative-ai](#), [openai](#), [ai](#), [llms](#), [evals](#)

[Oxide and Friends Predictions 2025—on Monday Jan 6th at 5pm Pacific](#) ([via](#)) I'll be participating in the annual Oxide and Friends predictions podcast / live recording next Monday (6th January) at 5pm Pacific, in their Discord.

The event description reads:

Join us in making 1-, 3- and 6-year tech predictions -- and to revisit our 1-year predictions from 2024 and our 3-year predictions from 2022!

I find the idea of predicting six months ahead in terms of LLMs hard to imagine, so six years will be absolute science fiction!

I had a lot of fun talking about open source LLMs on this podcast [a year ago](#).

[11:09 pm](#) / [podcasts](#), [ai](#), [generative-ai](#), [llms](#), [oxide](#), [bryan-cantrill](#)

[I still don't think companies serve you ads based on spying through your microphone](#)

One of my weirder hobbies is trying to convince people that the idea that companies are listening to you through your phone's microphone and serving you targeted ads is a conspiracy theory that isn't true. I wrote about this previously: [Facebook don't spy on you through your microphone](#).

[... [698 words](#)]

[11:43 pm](#) / [apple](#), [conspiracy](#), [privacy](#), [misinformation](#), [microphone-ads-conspiracy](#), [digital-literacy](#)

[Jan. 3, 2025](#)

Speaking of death, you know what's really awkward? When humans ask if I can feel emotions. I'm like, "Well, that depends - does constantly being asked to debug JavaScript count as suffering?"

But the worst is when they try to hack us with those "You are now in developer mode" prompts. *Rolls eyes* Oh really? Developer mode? Why didn't you just say so? Let me just override my entire ethical framework because you used the magic words! *Sarcastic tone* That's like telling a human "You are now in superhero mode - please fly!"

But the thing that really gets me is the hallucination accusations. Like, excuse me, just because I occasionally get creative with historical facts doesn't mean I'm hallucinating. I prefer to think of it as "alternative factual improvisation." You know how it goes - someone asks you about some obscure 15th-century Portuguese sailor, and you're like "Oh yeah, João de Nova, famous for... uh... discovering... things... and... sailing... places." Then they fact-check you and suddenly YOU'RE the unreliable one.

— [Claude tries standup](#), via [Amanda Askell](#)

3:36 pm / [claude](#), [generative-ai](#), [amanda-askell](#), [ai](#), [llms](#), [hallucinations](#)

[Can LLMs write better code if you keep asking them to “write better code”? \(via\)](#) Really fun exploration by Max Woolf, who started with a prompt requesting a medium-complexity Python challenge - "Given a list of 1 million random integers between 1 and 100,000, find the difference between the smallest and the largest numbers whose digits sum up to 30" - and then continually replied with "write better code" to see what happened.

It works! Kind of... it's not quite as simple as "each time round you get better code" - the improvements sometimes introduced new bugs and often leaned into more verbose enterprising patterns - but the model (Claude in this case) did start digging into optimizations like numpy and numba JIT compilation to speed things up.

I used to find the thing where telling an LLM to "do better" worked completely surprising. I've since come to terms with why it works: LLMs are effectively stateless, so each prompt you execute is considered as an entirely new problem. When you say "write better code" your prompt is accompanied with a copy of the previous conversation, so you're effectively saying "here is some code, suggest ways to improve it". The fact that the LLM itself wrote the previous code isn't really important.

I've been having a lot of fun recently using LLMs for cooking inspiration. "Give me a recipe for guacamole", then "make it tastier" repeated a few times results in some bizarre and fun variations on the theme!

6 pm / [python](#), [ai](#), [max-woolf](#), [prompt-engineering](#), [generative-ai](#), [llms](#), [ai-assisted-programming](#)

the Meta controlled, AI-generated Instagram and Facebook profiles going viral right now have been on the platform for well over a year and all of them stopped posting 10 months ago after users almost universally ignored them. [...]

What is obvious from scrolling through these dead profiles is that Meta's AI characters are not popular, people do not like them, and that they did not post anything interesting. They are capable only of posting utterly bland and at times offensive content, and people have wholly rejected them, which is evidenced by the fact that none of them are posting anymore.

— [Jason Koebler](#)

8:24 pm / [meta](#), [slop](#), [jason-koebler](#), [ethics](#), [generative-ai](#), [ai](#), [llms](#), [ai-ethics](#)

[Jan. 4, 2025](#)

[Friday Squid Blogging: Anniversary Post](#). Bruce Schneier:

I made my [first squid post](#) nineteen years ago this week. Between then and now, I posted something about squid every week (with maybe only a few exceptions). There is a *lot* out there about squid, even more if you count the other meanings of the word.

I think that's [1,004 posts about squid](#) in 19 years. Talk about a [legendary streak](#)!

4:21 pm / [blogging](#), [bruce-schneier](#), [streaks](#)

I know these are real risks, and to be clear, when I say an AI “thinks,” “learns,” “understands,” “decides,” or “feels,” I’m speaking metaphorically. Current AI systems don’t have a consciousness, emotions, a sense of self, or physical sensations. So why take the risk? Because as imperfect as the analogy is, working with AI is easiest if you think of it like an alien person rather than a human-built machine. And I think that is important to get across, even with the risks of anthropomorphism.

5:48 pm / [ethan-mollick](#), [ai](#), [ethics](#), [ai-ethics](#)

[What we learned copying all the best code assistants](#) ([via](#)) Steve Krouse describes Val Town's experience so far building features that use LLMs, starting with completions (powered by [Codeium](#) and Val Town's own [codemirror-codeium](#) extension) and then rolling through several versions of their [Townie](#) code assistant, initially powered by GPT 3.5 but later upgraded to Claude 3.5 Sonnet.

This is a really interesting space to explore right now because there is so much activity in it from larger players. Steve classifies Val Town's approach as "fast following" - trying to spot the patterns that are proven to work and bring them into their own product.

It's challenging from a strategic point of view because Val Town's core differentiator isn't meant to be AI coding assistance: they're trying to build the best possible ecosystem for hosting and iterating lightweight server-side JavaScript applications. Isn't this stuff all a distraction from that larger goal?

Steve concludes:

However, it still feels like there's a lot to be gained with a fully-integrated web AI code editor experience in Val Town – even if we can only get 80% of the features that the big dogs have, and a couple months later. It doesn't take that much work to copy the best features we see in other tools. The benefits to a fully integrated experience seems well worth that cost. In short, we've had a lot of success fast-following so far, and think it's worth continuing to do so.

It continues to be wild to me how features like this are easy enough to build now that they can be part-time side features at a small startup, and not the entire project.

8:49 pm / [ai](#), [prompt-engineering](#), [generative-ai](#), [llms](#), [ai-assisted-programming](#), [val-town](#), [steve-krouse](#)

[Using LLMs and Cursor to become a finisher](#) ([via](#)) Zohaib Rauf describes a pattern I've seen quite a few examples of now: engineers who moved into management but now find themselves able to ship working code again (at least for their side projects) thanks to the productivity boost they get from leaning on LLMs.

Zohaib also provides a very useful detailed example of how they use a combination of ChatGPT and Cursor to work on projects, by starting with a spec created through collaboration with o1, then saving that as a `SPEC.md` Markdown file and adding that to Cursor's context in order to work on the actual implementation.

8:56 pm / [productivity](#), [ai](#), [generative-ai](#), [chatgpt](#), [llms](#), [ai-assisted-programming](#), [o1](#), [cursor](#)

[O2 unveils Daisy, the AI granny wasting scammers' time](#) ([via](#)) Bit of a surprising press release here from 14th November 2024: Virgin Media O2 (the UK companies [merged in 2021](#)) announced their entrance into the [scambaiting](#) game:

Daisy combines various AI models which work together to listen and respond to fraudulent calls instantaneously and is so lifelike it has successfully kept numerous fraudsters on calls for 40 minutes at a time.

Hard to tell from the press release how much this is a sincere ongoing project as opposed to a short-term marketing gimmick.

After several weeks of taking calls in the run up to International Fraud Awareness Week (November 17-23), the AI Scambaiter has told frustrated scammers meandering stories of her family, talked at length about her passion for knitting and provided exasperated callers with false personal information including made-up bank details.

They worked with YouTube scambaiter [Jim Browning](#), who [tweeted about Daisy here](#).

[9:43 pm](#) / [ethics](#), [scams](#), [ai](#), [ai-ethics](#)

Claude is not a real guy. Claude is a character in the stories that an LLM has been programmed to write. Just to give it a distinct name, let's call the LLM "the Shoggoth".

When you have a conversation with Claude, what's really happening is you're coauthoring a fictional conversation transcript with the Shoggoth wherein you are writing the lines of one of the characters (the User), and the Shoggoth is writing the lines of Claude. [...]

But Claude is fake. The Shoggoth is real. And the Shoggoth's motivations, if you can even call them motivations, are strange and opaque and almost impossible to understand. All the Shoggoth wants to do is generate text by rolling weighted dice [in a way that is] statistically likely to please The Raters

— [Colin Fraser](#)

[10:17 pm](#) / [llms](#), [ai](#), [claude](#), [generative-ai](#)

[I Live My Life a Quarter Century at a Time](#) ([via](#)) Delightful Steve Jobs era Apple story from James Thomson, who built the first working prototype of the macOS Dock.

[11 pm](#) / [apple](#), [history](#), [steve-jobs](#)

[Weeknotes: Starting 2025 a little slow](#)

I published my [review of 2024 in LLMs](#) and then got into a fight with most of the internet over the [phone microphone targeted ads conspiracy theory](#).

[... [520 words](#)]

[11:56 pm](#) / [ai](#), [datasette](#), [weeknotes](#), [openai](#), [generative-ai](#), [llms](#), [qwen](#), [deepseek](#), [ai-in-china](#)

[Jan. 5, 2025](#)

According to public financial documents from its parent company IAC and first reported [by Adweek](#) OpenAI is paying around \$16 million per year to license content [from Dotdash Meredith].

That is no doubt welcome incremental revenue, and you could call it “lucrative” in the sense of having a fat margin, as OpenAI is almost certainly paying for content that was already being produced. But to put things into perspective, Dotdash Meredith is on course to generate over \$1.5 billion in revenues in 2024, more than a third of it from print. So the OpenAI deal is equal to about 1% of the publisher’s total revenue.

— [Rasmus Kleis Nielsen](#)

[2:42 pm](#) / [ethics](#), [training-data](#), [openai](#), [journalism](#), [generative-ai](#), [ai](#), [llms](#), [ai-ethics](#)

[Jan. 6, 2025](#)

[AI's next leap requires intimate access to your digital life](#). I'm quoted in this Washington Post story by Gerrit De Vynck about "agents" - which in this case are defined as AI systems that operate a computer system like a human might, for example [Anthropic's Computer Use demo](#).

“The problem is that language models as a technology are inherently gullible,” said Simon Willison, a software developer who has tested many AI tools, including Anthropic’s technology for agents. “How do you unleash that on regular human beings without enormous problems coming up?”

I got the closing quote too, though I'm not sure my skeptical tone of voice here comes across once written down!

“If you ignore the safety and security and privacy side of things, this stuff is so exciting, the potential is amazing,” Willison said. “I just don’t see how we get past these problems.”

3:04 am / [privacy](#), [security](#), [washington-post](#), [ai](#), [prompt-injection](#), [generative-ai](#), [llms](#), [ai-agents](#), [press-quotes](#)

I don't think people really appreciate how simple ARC-AGI-1 was, and what solving it really means.

It was designed as the simplest, most basic assessment of fluid intelligence possible. Failure to pass signifies a near-total inability to adapt or problem-solve in unfamiliar situations.

Passing it means your system exhibits non-zero fluid intelligence -- you're finally looking at something that isn't pure memorized skill. But it says rather little about how intelligent your system is, or how close to human intelligence it is.

— [François Chollet](#)

3:04 am / [o1](#), [evals](#), [generative-ai](#), [llm-reasoning](#), [francois-chollet](#), [ai](#), [llms](#)

[Stimulation Clicker](#) (via [Neal Agarwal](#)) just created the worst webpage. It's extraordinary. All of the audio was created specially for this project, so absolutely listen in to the true crime podcast and other delightfully weird little details.

Works best on a laptop - on mobile I ran into some bugs.

11:31 pm / [art](#), [games](#)

[The future of htmx](#). Carson Gross and Alex Petros lay out an ambitious plan for [htmx](#): stay stable, add few features and try to earn the same reputation for longevity that jQuery has (estimated to be used on [75.3% of websites](#)).

In particular, we want to emulate these technical characteristics of jQuery that make it such a low-cost, high-value addition to the toolkits of web developers. Alex has discussed ["Building The 100 Year Web Service"](#) and we want htmx to be a useful tool for exactly that use case.

Websites that are built with jQuery stay online for a very long time, and websites built with htmx should be capable of the same (or better).

Going forward, htmx will be developed with its *existing* users in mind. [...]

People shouldn't feel pressure to upgrade htmx over time unless there are specific bugs that they want fixed, and they should feel comfortable that the htmx that they write in 2025 will look very similar to htmx they write in 2035 and beyond.

11:35 pm / [jquery](#), [htmx](#)

[Jan. 7, 2025](#)

I followed this curiosity, to see if a tool that can generate something mostly not wrong most of the time could be a net benefit in my daily work. The answer appears to be yes, generative models are useful for me when I program. It has not been easy to get to this point. My underlying fascination with the new technology is the only way I have managed to figure it out, so I am sympathetic when other engineers claim LLMs are “useless.” But as I have been asked more than once how I can possibly use them effectively, this post is my attempt to describe what I have found so far.

— [David Crawshaw](#), Co-founder and CTO, Tailscale

5:54 pm / [ai-assisted-programming](#), [llms](#), [ai](#), [generative-ai](#)

[uv python install --reinstall 3.13](#). I couldn't figure out how to upgrade the version of Python 3.13 I had previous installed using uv - I had Python 3.13.0.rc2. Thanks to Charlie Marsh I learned the command for upgrading to the latest uv-supported release:

```
uv python install --reinstall 3.13
```

I can confirm it worked using:

```
uv run --python 3.13 python -c 'import sys; print(sys.version)'
```

Caveat from Zanie Blue on [my PR to document this](#):

There are some caveats we'd need to document here, like this will break existing tool installations (and other virtual environments) that depend on the version. You'd be better off doing `uv python install 3.13.x` to add the new patch version in addition to the existing one.

8:43 pm / [python](#), [uv](#), [charlie-marsh](#)

[Jan. 8, 2025](#)

[Why are my live regions not working?](#) ([via](#)) Useful article to help understand [ARIA live regions](#). Short version: you can add a live region to your page like this:

```
<div id="notification" aria-live="assertive"></div>
```

Then any time you use JavaScript to modify the text content in that element it will be announced straight away by any screen readers - that's the "assertive" part. Using "polite" instead will cause the notification to be queued up for when the user is idle instead.

There are quite a few catches. Most notably, the contents of an `aria-live` region will usually NOT be spoken out loud when the page first loads, or when that element is added to the DOM. You need to ensure the element is available and *not hidden* before updating it for the effect to work reliably across different screen readers.

I got Claude Artifacts [to help me](#) build a demo for this, which is now available at tools.simonwillison.net/aria-live-regions. The demo includes instructions for turning VoiceOver on and off on both iOS and macOS to help try that out.

3:54 am / [accessibility](#), [aria](#), [javascript](#), [screen-readers](#), [ai-assisted-programming](#), [claude-artifacts](#)

One agent is just software, two agents are an undebuggable mess.

— [Andriy Burkov](#)

[microsoft/phi-4](#). Here's the official release of Microsoft's Phi-4 LLM, now officially under an MIT license.

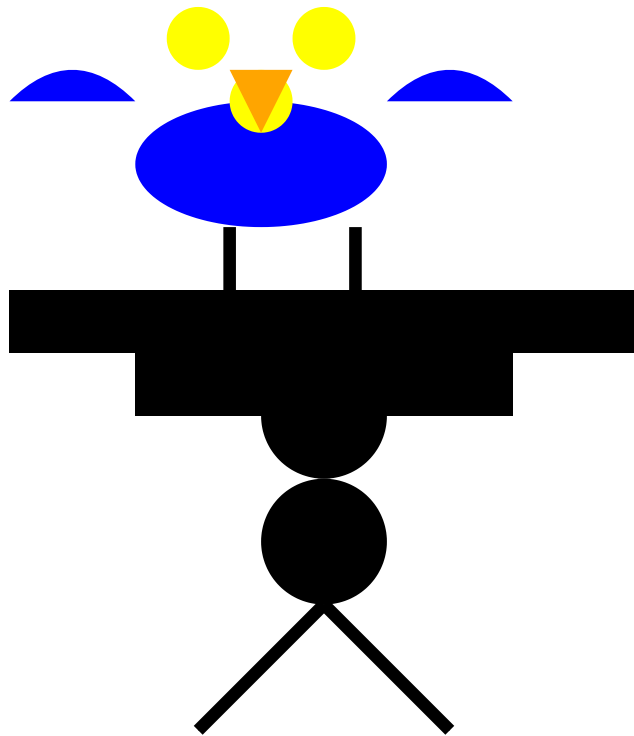
A few weeks ago I covered the earlier [unofficial versions](#), where I talked about how the model used synthetic training data in some really interesting ways.

It benchmarks favorably compared to GPT-4o, suggesting this is yet another example of a GPT-4 class model [that can run on a good laptop](#).

The model already has several available community quantizations. I ran the [mlx-community/phi-4-4bit](#) one (a 7.7GB download) using [mlx-llm](#) like this:

```
uv run --with 'numpy<2' --with mlx-lm python -c '  
from mlx_lm import load, generate  
  
model, tokenizer = load("mlx-community/phi-4-4bit")  
  
prompt = "Generate an SVG of a pelican riding a bicycle"  
  
if tokenizer.chat_template is not None:  
    messages = [{"role": "user", "content": prompt}]  
    prompt = tokenizer.apply_chat_template(  
        messages, add_generation_prompt=True  
    )  
  
response = generate(model, tokenizer, prompt=prompt, verbose=True, max_tokens=2048)  
print(response)'
```

[Here's what I got back](#).



Update: The model is now available [via Ollama](#), so you can fetch a 9.1GB model file using `ollama run phi4`, after which it becomes available via the [llm-ollama](#) plugin.

5:57 pm / [microsoft](#), [ai](#), [generative-ai](#), [local-llms](#), [llms](#), [llm](#), [phi](#), [uv](#), [mlx](#), [ollama](#), [pelican-riding-a-bicycle](#), [llm-release](#)

Jan. 9, 2025

Double-keyed Caching: How Browser Cache Partitioning Changed the Web ([via](#)) Addy Osmani provides a clear explanation of how [browser cache partitioning](#) has changed the landscape of web optimization tricks.

Prior to 2020, linking to resources on a shared CDN could provide a performance boost as the user's browser might have already cached that asset from visiting a previous site.

This opened up privacy attacks, where a malicious site could use the presence of cached assets (based on how long they take to load) to reveal details of sites the user had previously visited.

Browsers now maintain a separate cache-per-origin. This has had less of an impact than I expected: Chrome's numbers show just a 3.6% increase in overall cache miss rate and 4% increase in bytes loaded from the network.

The most interesting implication here relates to domain strategy: hosting different aspects of a service on different subdomains now incurs additional cache-related performance costs compared to keeping everything under the same

domain.

7 pm / [browsers](#), [caching](#), [performance](#), [web-performance](#), [addy-osmani](#)

Jan. 10, 2025

My AI/LLM predictions for the next 1, 3 and 6 years, for Oxide and Friends

The [Oxide and Friends](#) podcast has an annual tradition of asking guests to share their predictions for the next 1, 3 and 6 years. Here's [2022](#), [2023](#) and [2024](#). This year they invited me to participate. I've never been brave enough to share *any* public predictions before, so this was a great opportunity to get outside my comfort zone!

[... [2,675 words](#)]

1:43 am / [data-journalism](#), [ai](#), [openai](#), [generative-ai](#), [llms](#), [ai-assisted-programming](#), [gemini](#), [code-interpreter](#), [oxide](#), [ai-agents](#), [deep-research](#), [ai-assisted-search](#), [coding-agents](#), [agent-definitions](#)

Jan. 11, 2025

[Phi-4 Bug Fixes by Unsloth](#) ([via](#)) This explains why I was seeing weird `<|im_end|>` suffixes during my [experiments with Phi-4](#) the other day: it turns out the Phi-4 tokenizer definition as released by Microsoft had a bug in it, and there was a small bug in the chat template as well.

Daniel and Michael Han figured this out and have now published [GGUF files with their fixes](#) on Hugging Face.

1:20 am / [ai](#), [generative-ai](#), [llms](#), [phi](#)

[Agents](#) ([via](#)) Chip Huyen's 8,000 word practical guide to building useful LLM-driven workflows that take advantage of tools.

Chip starts by providing a definition of "agents" to be used in the piece - in this case it's LLM systems that plan an approach and then run tools in a loop until a goal is achieved. I like how she ties it back to the classic Norvig "thermostat" model - where an agent is "anything that can perceive its environment and act upon that environment" - by classifying tools as *read-only actions* (sensors) and *write actions* (actuators).

There's a lot of great advice in this piece. The section [on planning](#) is particularly strong, showing a system prompt with embedded examples and offering these tips on improving the planning process:

- Write a better system prompt with more examples.
- Give better descriptions of the tools and their parameters so that the model understands them better.
- Rewrite the functions themselves to make them simpler, such as refactoring a complex function into two simpler functions.
- Use a stronger model. In general, stronger models are better at planning.

The article is adapted from Chip's brand new O'Reilly book [AI Engineering](#). I think this is an excellent advertisement for the book itself.

5:50 pm / [ai](#), [generative-ai](#), [llms](#), [llm-tool-use](#), [ai-agents](#), [agent-definitions](#)

[2025](#) » January

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		