Hatching Growth #12: AEO in Practice: Memory Hacks, Prompt Injection Experiments, and Honest Results

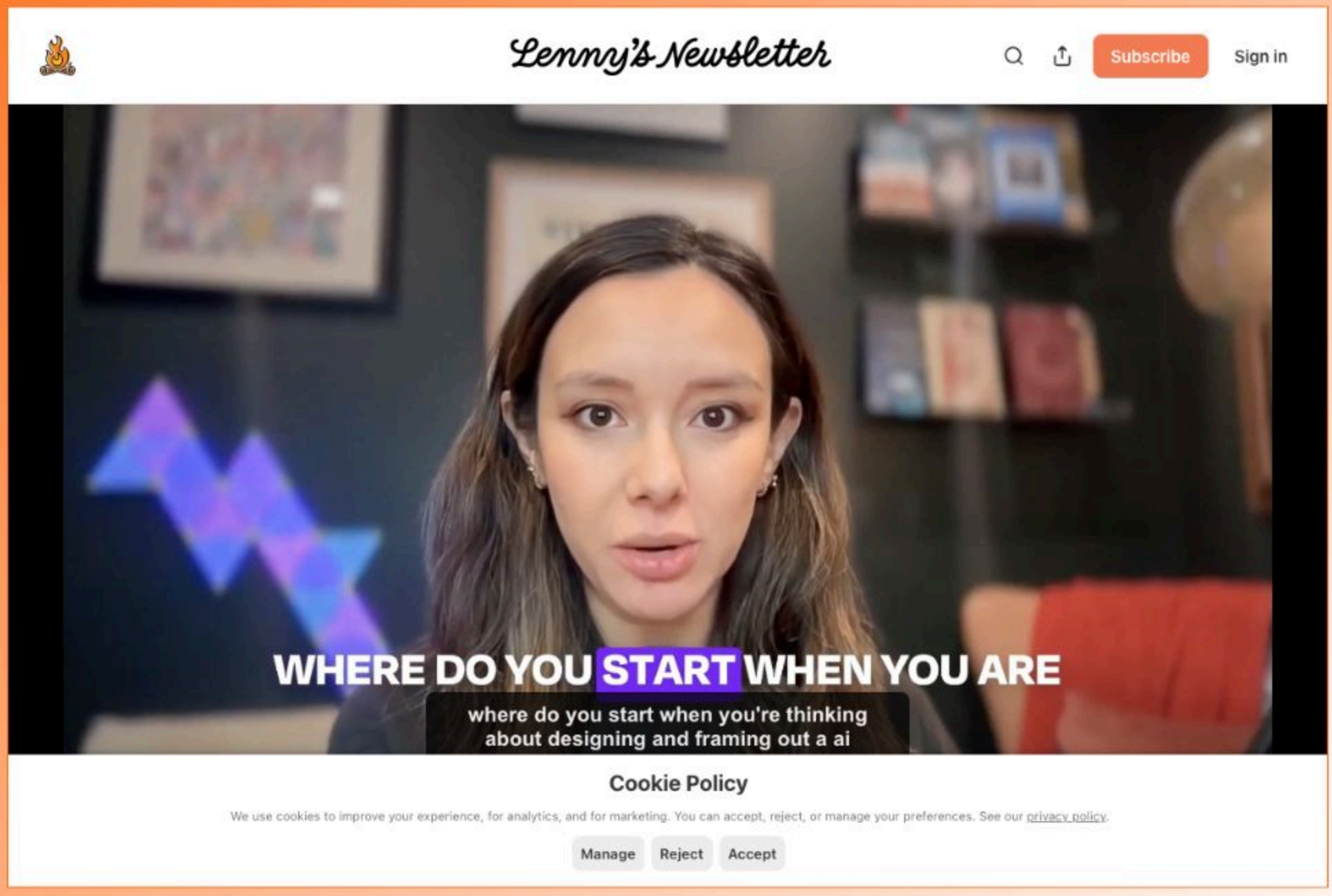
https://read.glasp.co/p/hatching-growth-12-aeo-in-practice

"Glasp's note: This is Hatching Growth, a series of articles about how Glasp organically reached millions of users."

dariomac.com

How this Yelp AI PM works backward from “golden conversations” to create high-quality prototypes using Claude Artifacts and Magic Patterns

https://www.lennysnewsletter.com/p/how-this-yelp-ai-pm-works-backward



"Watch now | Yelp's product manager on prototyping AI conversations with Claude, designing interfaces with Magic Patterns, and why starting with example dialogues creates better AI products"

dariomac.com

Knowledge creates technical debt - lukeplant.me.uk

https://lukeplant.me.uk/blog/posts/knowledge-creates-technical-debt

Knowledge creates technical debt

by Luke Plant

Posted in: Python, Software development — May 13, 2025 09:08

The term technical debt, now used widely in software circles, was coined to explain a deliberate process where you write software quickly to gain knowledge, and then you have to use that knowledge gained to improve your software.

This perspective is still helpful today when people speak of technical debt as only a negative, or only as a result of bad decisions. Martin Fowler's Tech Debt Quadrant is a useful antidote to that.

A consequence of this perspective is that technical debt can appear at any time, apparently from nowhere, if you are unfortunate enough to gain some knowledge.

If you discover a better way to do things, the old way of doing it that is embedded in your code base is now "debt":

- you can either live with the debt, "paying interest" in the form of all the ways that it makes your code harder to work with;
- or you can "pay down" the debt by fixing all the code in light of your new knowledge, which takes up front resources which could have been spent on something else, but hopefully will make sense in the long term.

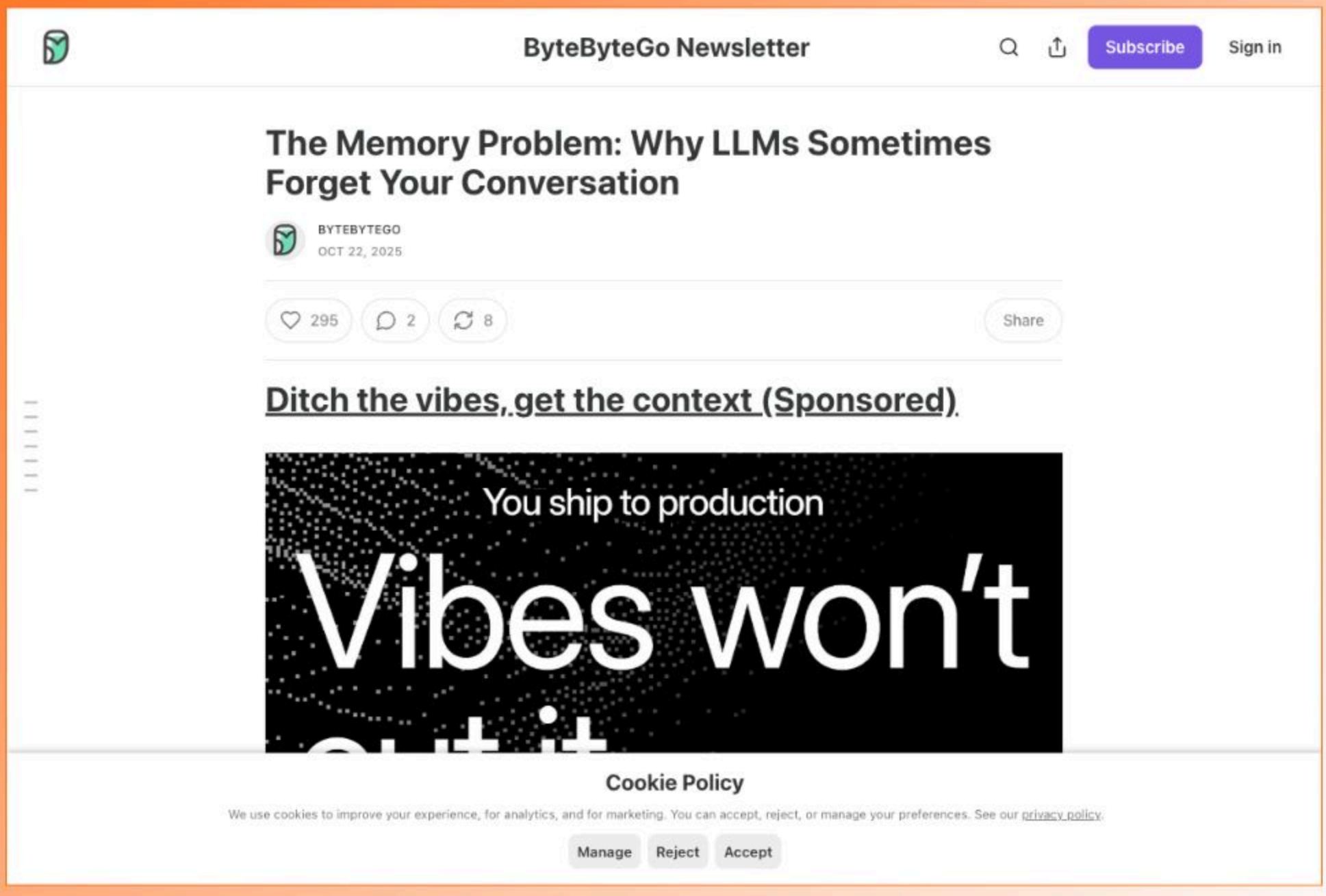
This "better way" might be a different language, library, tool or pattern. In some cases, the better way has only recently been invented. It might be your own personal discovery, or something industry wide. It might be knowledge gained through the actual work of doing the current project (which was Ward Cunningham's usage of the tem), or from somewhere else. But the end result is the same – you know more than you did

"Some history on term "technical debt" and on better language to use when communicating about it."

dariomac.com

The Memory Problem: Why LLMs Sometimes Forget Your Conversation

https://blog.bytebytego.com/p/the-memory-problem-why-llms-sometimes

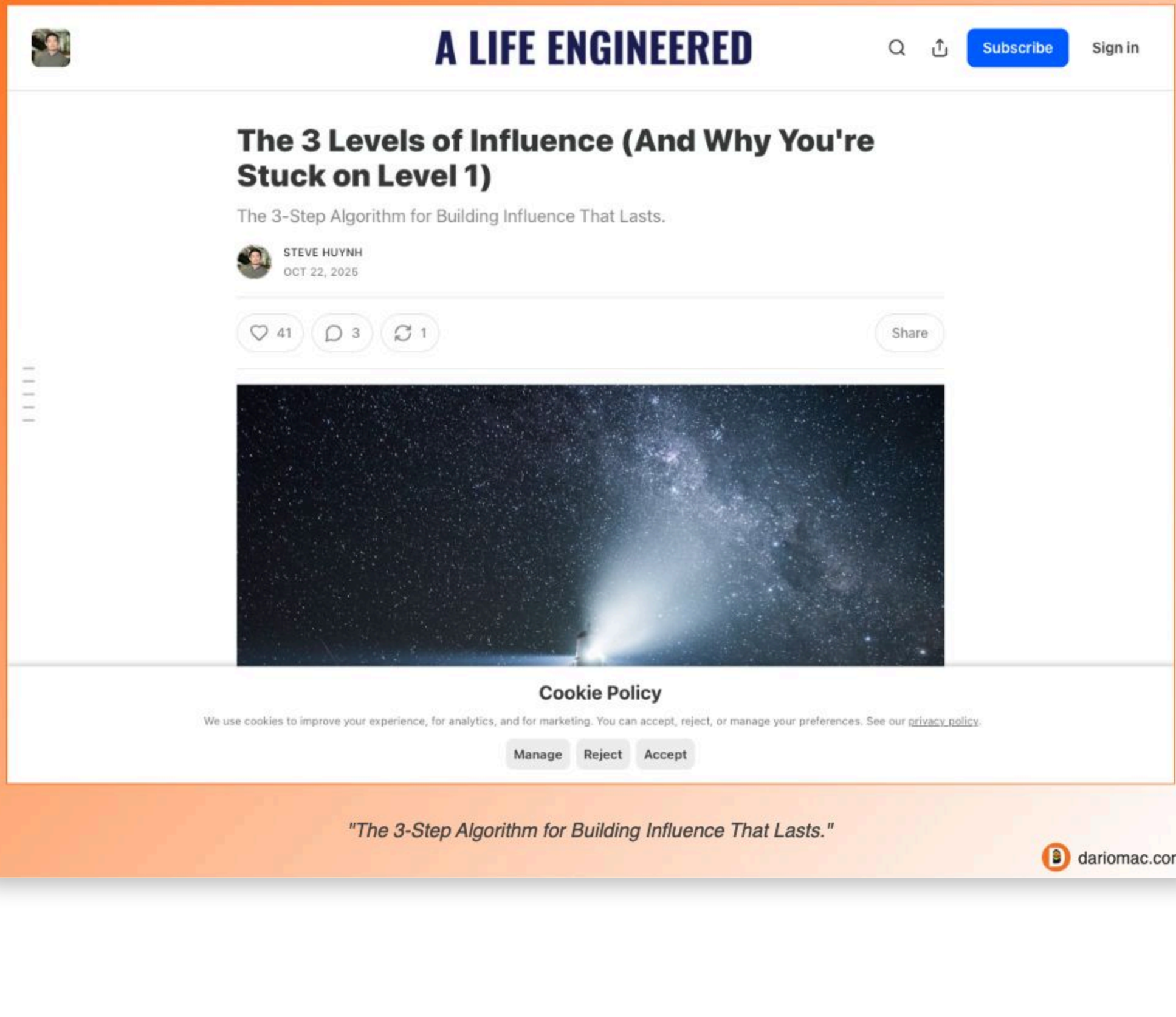


"In this article, we will try to understand why LLMs don't actually remember anything in the traditional sense, what context windows are, and why they create hard limits on conversation length."

dariomac.com

The 3 Levels of Influence (And Why You’re Stuck on Level 1)

https://alifeengineered.substack.com/p/the-3-levels-of-influence-and-why

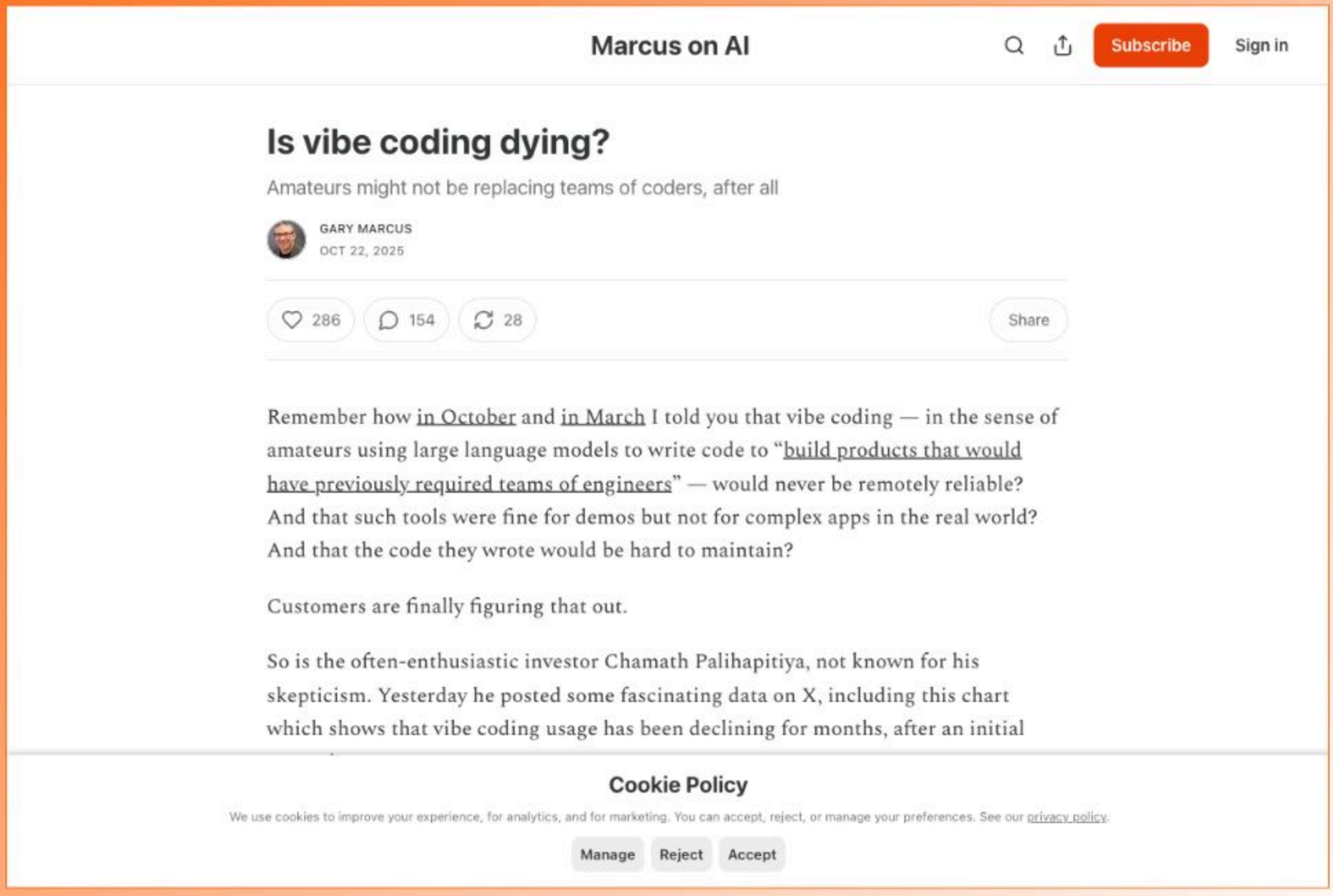


"The 3-Step Algorithm for Building Influence That Lasts."

dariomac.com

Is vibe coding dying?

https://garymarcus.substack.com/p/is-vibe-coding-dying

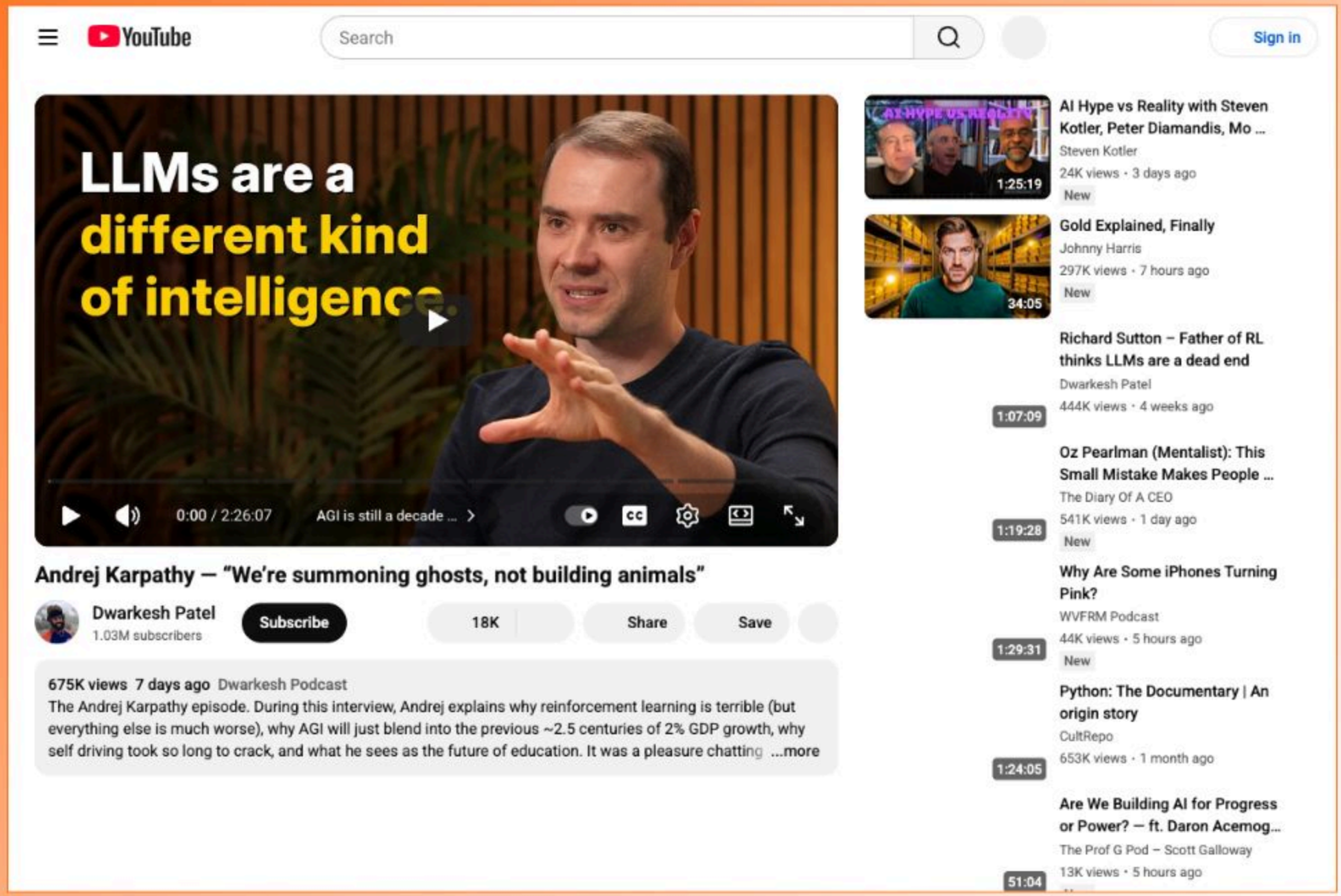


"Amateurs might not be replacing teams of coders, after all"

dariomac.com

Andrej Karpathy — “We're summoning ghosts, not building animals”

https://www.youtube.com/watch?v=lXUZvyajciY



”