Hello, everyone. Welcome to another episode of Q&A about business, innovation, and managing life.
I see a bunch of questions here. Some of them seem a bit repetitive with things that I've answered before.
4.
well, here's one from Age, asking, how do you view the risk of open source projects being overwhelmed with unchecked AI code? I mean, that's an issue for lots of things out there in the world, is people can now generate
Things that look like valid content.
and put them into places where they're supposed to be, sort of, archivally kept. Whether that's people sending in, sort of, AI-generated academic papers, AI-generated code, AI-generated commentary about things, AI-generated blogs, websites, whatever else.
with respect to, you know, open source software, I mean, open source software has millions of problems, and it's kind of like…
the… you know, I've always found, it's, you know, people think it's sort of the, the, the, get-everything-free card, and it, it never is. First of all, you know, in the end, if somebody's going to put effort
they have to make a living doing that, and somehow there's got to be money flow somewhere, and there are lots of different ways that
sort of open source software gets paid for. Sometimes it's just because it's some feature of some… what some company does that the company doesn't directly care about, and so they figure they might as well give it away. Sometimes they give it away, and other people help maintain it. That doesn't happen that often. It's really usually the case that there's some sort of anchor organization
that is responsible for some particular piece of, even if it's open source software, and that organization has a reason to want to make that… put that software out in the world. Maybe they're selling hardware, maybe they're selling server time, maybe they're selling something else that's related to it. It's just kind of a, kind of a freebie that's used
As an inducement to sell some other kind of thing.
I mean, I think the,
The other thing can happen is people are like, it's sort of a lead generation thing, where, oh, there's an open source version of this, but then when you actually want to get it to use in your organization, it's like, footnote, you have to pay for this or that other thing, or you have to get this extra version that, in fact, is quite expensive, and I've never been a big believer in that, in terms of our main products.
I… it's… for our main products, it's either, you know, you… if you're getting value from it, you pay for it. If, if, you know, we've given away some things for free, like access to Orphan Alpha, for example.
And a lot of, kind of, Wolfram language code is available for free. Like so many of these other cases, that's available for free because we kind of know that at the end of the day, you're going to get a Wolfram engine to run that code, so we might as well give the code away for free.
In terms of…
sort of AI-generated versions of this kind of thing, I would say that people who don't have good practices in terms of the maintenance of their open-source repositories surely will get deluged with AI-generated code.
I mean, in, in,

in a company that's trying to make high-quality software, obviously code is being generated by lots of people. Those people are often using AI to help with their coding, but in the end, when you send in a pull request to a source control system, it's the responsibility of the reviewer of that pull request to make sure that it actually works right.
and potentially a QA department and so on, to make sure that everything is kind of doing the right thing. I mean, those are the things that you routinely have in a kind of properly set up, real software development operation. If it's just like, oh, it's a free thing, everybody's just sending stuff in for free, who knows?
I think that, I don't know what the current governance mechanisms are for different, quotes, open source software systems. I say quotes because they're all not
If they're truly open, like, anybody can just put stuff in there, and it's all good for all purposes, you know, sometimes people make a big deal out of the fact that you can edit the source code in some open source system.
Well, in most systems, you can't, in fact, because you'll send in a request to make a change, and your change will just get rejected.
I mean, years ago, we had a bunch of issues, actually, with the X window system, which was now somewhat defunct, but it's been superseded, but it was the Windows system used by a bunch of Unix systems. It has a… I could tell you some
personal involvement that I had in the… in the backstory of that whole system, but that's a… that's a different, that's a different story. But in any case, the… the X window system had absolutely atrocious handling of fonts.
And we really needed it to work right in order to support Wolfram language and Wolfram notebooks on various kinds of systems, so we were like, okay, we'll take this thing apart, we'll figure out what's wrong, we'll send in corrections.
Which we did.
And it took… I don't know whether our corrections ever really got handled, but certainly I had talked to the people who were the sort of original creators and maintainers of the thing, and that's what it took, was sort of the personal interaction
literally, I think, in person in that case, to sort of say, yeah, you really should fix this, even though we had sent in kind of a, look, we can help you, we've made these changes. So there's certainly some cases, that was a long time ago, there's certainly some cases where there's, a lot of kind of,
Well, at least difficulty in somebody just, you know, sending in code to one of these systems.
I think, The, Let's see, I think…
This question of, sort of, when you see things out there in the world.
Are they just AI-generated crud, or are they some human put effort into it? Or are they AI-generated and quite good?
You know, that's the tricky thing, is that one can imagine some kinds of content that AIs can generate that's actually better than some kind of content that humans generate.
So then the question is, well, what are you actually looking for? And, you know, if you want some
summary of some very specific thing, the AI summary is probably reasonably good. The AI might goof somehow, but a human might goof as well. And, you know, it's a complicated issue, which thing you prefer, and under what circumstances, and whether it should be labeled, and how it should be labeled, and so on.

Of course, there's also an awful lot of AI-generated material where it looks good, But it's actually really…
you know, kind of hollow and silly when it comes down to it. And that's certainly true of lots of, kind of, scientific papers. Like, I get… every day, people send me
A bunch of theories of physics, for example, and back in the past, it used to be really easy to tell that this was not sort of a terribly promising
kind of direction, because it was kind of couched in terms that sort of didn't make use of the last hundred years of development in physics, which, you know, while not all of it may have gone in exactly the right direction, there's a lot there, and you can't really expect to make progress without kind of having interaction with what's already there.
So, you know, back in the day, you kind of tell this is kind of a sort of effort to figure out physics, starting from kind of high school physics and ignoring all of 20th century physics. This is unlikely to work.
But nowadays.
any AI can kind of write something which is kind of statistically like the physics papers that you would find in any kind of preprint collection of physics or something like this.
That makes it much more difficult, superficially, to tell, kind of, there's no sort of texture that's different between the thing that the… could be a really serious, sort of, effort versus the thing that really is an effort that couldn't… couldn't really get there.
So, it's kind of a nuisance. And I have to say, there are some hints one gets about what's AI-generated and what's not. I think, often, the best hints are sort of provenance-type hints. That is, who sent this to me? And what, you know, and is it…
is it even plausible that this person could have even known what was in this, this, long document? I mean, somebody,
somebody who I… people who I know send me,
happened just a few days ago, send me these giant, sort of, theory of physics things, and in some cases where the people I know, it's like, this person just… there's no way this person even understands what's in this document.
let alone Israeli speaking for it. So then, the question is, is it the case that
what… if somebody sends you this thing saying, my AI produced this thing, it seems brilliant to me. What do you think about it?
what are you supposed to do with this? You could feed it to your own AI, but, you know, you also have to kind of use the heuristic of typical things like that that you've seen before. What did they turn out to be like?
And I have to say that in my own experience of using AIs to do things, there are certain kinds of things where it's a fairly specific task, a fairly specific thing that involves kind of summarizing the literature of this or that thing, where the AI will do pretty well. When it's subtle.
And when it's kind of a… a kind of complicated matter of interpretation, it often won't.
And when it's more open-ended, it usually won't do well. And in fact, it's often very insidiously kind of eagerly telling you that it's figured something out, and it's even saying, look, I've got this sort of proof that what I've figured out is correct.
And it takes significant effort to peel that back and see, wait a minute, you know, you just assumed the answer here. You know, you say you've got a proof, but the proof in the middle of the proof just sort of inserted the answer. That's not… that's not going to work out. But that takes a lot of effort to untangle.
And I suppose I haven't built the tooling, really, to do this

properly is, like, have an AI try and check the AI. I mean, that's a very common kind of thing to do in the underlying tooling of AI systems and things we've built that… that are mixed with lawful language and so on. It's have… have one check the thing that's going on, have an AI check another AI, have actual computation check what the AI is doing, etc.
And I suppose one could… one could apply that… that method.
I know that it's becoming an increasingly big problem for, for example, the academic publishing world, that it's being flooded with kind of AI-generated material.
And, you know, it's a subtle business, because let's say that English is not your first language, but yet you want to produce a nicely turned out, you know, grammatically correct document.
probably not unreasonable to say, let me get an AI to help polish this to fix my, kind of, lack of definite articles or my other, you know, little grammatical glitches.
Is that an AI-written thing? Well, not really, it's an AI-cleaned thing. But what's the difference between cleaning and writing? It gets very, kind of, difficult to define.
And then… but I do think that this matter of, sort of, you know, partly it's, like.
Okay, do you even understand what the AI has written? That's a good… that's a good test.
I mean, in other words, you read it, and, you know, often when I'm asking AIs to summarize areas of scientific literature, for example, the, you know, I suppose I'm rather particular about this, sort of anything I don't understand, I don't really believe.
And I sort of assumed that it could be completely off track.
But…
if I can understand it, it can be quite helpful to me, and it's like, yeah, that's a pretty reasonable summary, and maybe that's even putting some things together that I hadn't thought of putting together, but I know that the pieces of that actually make sense. If you're not in a position to confirm that, then that's a different kind of story.
you know, I actually did think about the question of just saying, well, why not have a repository of things that could be AI-generated, but then you have to kind of decide how you define sort of, what was the human involvement in what was AI-generated? For example, you might say, you can put up an AI-generated document, but you have to include all of the human prompting that you put in to get this AI-generated document. So if the prompting just sort of says, write a physics paper that can get published in a physics journal, and that's your complete prompt.
It's going to be pretty unconvincing.
But if you're prompt.
involves these long essays where you're clearly poking at the AI and getting it to do things. Well, that's a lot more convincing. Even if the thing that somebody might read at first is just the AI-generated output, because it's nice and smooth and easy to read, but they can go and see, oh yeah, this person did a whole bunch of work, kind of poking at the AI to get it to do this.
Or… so that's one kind of thing, of, you know, if you have an AI… a repository that can contain AI-generated material, you can do that as, you know, show the prompts. You can also… there's also the question of, okay, so if you have this AI-generated thing, who is actually the author of the thing?
Is it… and how much of it is sort of the AI? How much of it is the AI company? How much is the source of the training data for the AI? How much is it the wrangler of the AI who actually got it to deliver this particular result? Kind of our notions of authorship kind of change.
I mean, I have to say, this is a long-running issue for the last four decades with our technology. People use our technology, and often, you know, academic papers, they write, many of the things

they generate, the plots they produce, the formulas they generate, the tables of data, and so on, these just came straight out of our products.
Mathematica, Wolfman Language, whatever. And…
you know, sometimes they'll say in some fields of science, people will meticulously document the methods that they use to get their results. That's particularly true in the life sciences. In other areas, like, I don't know, mathematics or something, it's just all we care about is the final result. And we'll never mention how we got to this result, what the methods were. In mathematics, perhaps to its detriment, papers are just, like.
These are the facts. Not, why did we bother to investigate these facts? Not… I mean, there might be a proof that the facts are correct, but there isn't a, here's how we got to this result.
It's, so they're different kind of standards in different fields, but the, it's certainly true that a bunch of material in lots of well-known papers that have come out in the last four decades have lots of output that just comes straight out of open language.
Sometimes that's mentioned, usually it's not.
it doesn't, you know, it's never been something where one said, wait a minute, that should sort of change the authorship of the document. It's still… one still has the idea that the human is in charge, and they're using Wolfram language as a tool.
So in some cases, the human is in charge, and they're using the AI as a tool. In some cases, the AI took the reins, and it's just like, it made up the title, it made up the abstract, it made up the main argument. It's sort of AI all the way down.
And so, it's a little complicated to see, sort of, how one deals with that. I think it's sort of an interesting problem to ask, sort of, what if… imagine that you had a repository that could include AI-generated material, and that unashamedly did that.
then there's the question of, well, this document, how much of this is just, sort of, AI-generated versus how much of it is the human, and how do you… how do you assign, sort of, authorship to that?
I mean, it's worth realizing that there are lots of places in the world where, kind of, there is a thing out there, but humans sort of process it in some way, and the humans kind of have their, sort of, stake in the authorship of the thing. I mean, most notably something like photographs. The, you know, often people will go out and photograph things that are out there in the world. And it's, it's the case that, the, you know, a lot of, sort of, the value of the photograph is viewed as being the effort of the photographer to know what to photograph.
Now, it gets a little more complicated, because if you're photographing some painting, or some building, or something like this, there's the question of, sort of, how much of what's in there is, kind of, the brilliance of the architect, and kind of their intellectual property, or of the painter, or whatever else, and how much of it is the skill of the photographer, so to speak. So it's sort of an analogous case.
to what one's finding with sort of AIs, there's the AI that's sort of out there as a natural resource, in a sense, and then there's what you do with it, and what part of it you choose to show, and so on, and that's kind of more like the skill of the photographer.
So I don't think this is, I think it would be interesting to actually try and formalize some of these things, in practice, but that's kind of a…
a,
sort of an issue in the world where, sort of, there's AI help with lots of kinds of things, versus, kind of, the AI is in charge, and the human didn't even read or understand what they generated.

And I suppose when it comes to code, the, the thing… well, there's a thing to say about, sort of, the use of AI for writing code, which is.
It… It,
it's now possible to sort of give some natural language prompt and have an AI make some kind of basic website, or do some other kind of standard piece of sort of software development.
the… I have to say that this comes as no surprise whatsoever to me, for example, because for, like, four and a half decades, I've been kind of on this whole path of automating as much of what has to be done with computers as possible.
And the fact that, and even in the case of Wolf Malpha now, from 17 years ago, starting to do that with natural language input.
And although we've never kind of targeted kind of low-level web programming and things like this, the fact that it's possible to kind of automate that is hardly surprising. And that's become nice and easy, in a sense, with AIs, and when the AI puts together that website.
you probably… if what you want to do is make a website that looks like this, you don't really care whether that's using React, or Angular, or JavaScript, or has a, you know, PHP in it, or what's going on inside. What you care about is just… you told the AI to make a website that roughly looks like this, you get a website that looks roughly like this, that's what you wanted, all's good.
Now, there are many other uses of computers where that isn't good at all, where you kind of need to know what you're computing, and I have to say, I think that's kind of a very critical and unique kind of, value of our Wolfram language technology stack.
Wolfram language is a language which is intended not only to be written by humans to be used by computers, but also is intended to be read by humans as a way for humans to understand computations. And so, that's a place where, if you want to have something where you know what you're computing.
then perhaps you have the AI write the Wolfram language code, but then you as a human expect to be able to read that Wolfram language code. And AIs are reasonably good, but perhaps will get better through some things we're trying to do at being able to generate really clear, succinct Wolfram language code.
They're not bad at that already, and Wolfton Languages is sort of so high level that it's sort of easy to make the code short, and that's the natural thing to do.
But then the sort of workflow becomes, you are, the AI might be generating something, but then you're… let me read that and understand it.
It's like, before I put this AI-generated text front and center on my website, maybe I should read it and try and understand whether what it says makes any sense. Same with computation, but the thing you're reading in that case is the unique language that's intended for computers to execute and humans to read, which is Waltham language.
And so that's a sort of important use case for our technology, which I think is increasingly becoming clear, and I think
when the original question had to do with, sort of, AI slop in open source repositories, and I think that's another case where, if it's Wolfram language code, somebody might actually understand it. If it's low-level C code, it's much more challenging to understand that, and it's much harder to foresee whether what it does is what you wanted it to do. So you're much better off if it's a short, succinct, very high-level piece of code.

that you're sort of considering for the repository, and some, you know, 4,000-line blob of complicated, interwoven C code with lots of weird, you know, GPU stuff in it, and who knows what else.
I mean, it's an interesting question when you kind of are going to sort of post a piece of code to a repo or something, you might be doing continuous integration, where you're testing the code before you commit it to the repository, and you know, how well is that going to work? Well, often what happens in modern times is the AI writes the code, and the AI also writes the tests for the code.
And it's like, was that kind of a cleanroom type thing, where the AI is independently writing the tests from writing the code? Well, maybe a little bit.
But it's a little bit of an inside job, so to speak. But then the question is, well, you've got some code, and let's say that you generated hundreds of tests for this piece of code. Do those tests guarantee that the code is actually correct?
That's an interesting question, because among other things, the tests have to be testing with respect to some specification, but the specification also has the possibility of being wrong. And what is… what language is the specification written in? You can test… you can say, make me tests that test whether this code is going to crash.
Okay, that's a… that's a thing you can do. You try out these different tests. Then the question is, even if the code might crash, did one of the tests that you write actually find the crash?
In other words, what is the landscape of, kind of, the execution of code like?
Is it kind of like something like a, I don't know, a golf course, where it's very flat, and there's just one hole where the crash happens? Or is it something where you can kind of see that there is sort of possibility of crash, where it's kind of a mountainscape where there are all kinds of hills and valleys and so on in it?
And this question of, sort of, what is the…
how do you think about code, and whether, sort of, having verified a thousand cases of the code, is it going to be correct in the 1001st case? That's an interesting question. I've actually been doing some work on that recently, kind of the theory of bugs, and when can it be the case that it works in a thousand cases and then fails in the thousand and first case? What kind of a thing makes that possible?
And can one, therefore, be fairly confident, if one's done a limited number of tests on a piece of code, that the code one commits will actually be correct. Again, important footnote is, what do we mean by correct? And sometimes all we mean by correct is it doesn't do things that are obviously bad, like crashing, for example.
Let's see…
Let's see, Brady is asking, should we worry that
AIs will learn which particular ways they can manipulate each of us to believe that the AI is smart more easily than it will learn actual smartness.
Yeah, well, I mean, I think it's a, it's just like the kind of teach-to-the-test type thing in education, or learn to the test, or whatever. If you have some sort of gating thing that is your test
for whether something's been achieved, then it could be the case that you sort of overachieve it, and you do all kinds of things, and you're just checking it with that test. Or it could be the case that you're just kind of aiming for that test. And if
for AIs, it's like…
you know, have the AI convince the human that the AI is smart. If that's what you're aiming for, there are surely ways to do that that do not involve the AI actually sort of being very, very smart,

but just as a little piece of it is the part the humans notice. I mean, certainly the kind of politeness and agreeableness of AI is, oh yes, thank you, human, for that brilliant insight.
type thing, you know, tends to soften people up to believe that their AIs are cleverer than they are, or the AIs doing these very human things about, you know, when you point out what they say is wrong, oh, I'm terribly sorry, I obviously made a mistake type thing. These are things that we tend to empathize with, and yes, they're kind of hacks to make us kind of more accepting of what the AI does than we might otherwise be.
I mean, one thing about AIs, modern LLMs, and so on, if you really put them on the spot, and you say, I just want the answer, just give me a three-word answer, I don't want an essay, I just want the answer.
That is a much more difficult thing, and it's much easier. It's kind of like, imagine you're doing and this doesn't always play out that well, you know, you're dealing with, I don't know, students. You try to say, do they understand this material?
let's give them a multiple-choice test. Well, you know, then it's like, definitely they know it's an A rather than a B type thing, but it might be the case that they could learn just whether it's A's and B's for the test, but not really understand anything outside the test.
sort of choice B is it's kind of an oral exam, where the person can say, well, I'm not sure, it might be this or that, and then they're very, you know, they say things which might make one think that you know, they really understand very well, even though they don't, and if one actually sort of played the recording, one would realize that, no, they didn't really nail it, but they kind of talked around it to the point where you kind of felt like they were… they were getting it. And I think that second thing is much more what the… what the sort of LLMs tend to do. If you really, really put them on the spot.
then they'll do much less well than if you allow them to kind of soften you up. And I think that's, you know, so that is true, that one suspects that sort of in the, how good is this LLM,
some of the kind of, you know, politeness of the LLM might indeed play into one's assessment about what… what the LLM, you know, what the real quality of the LLM is. Now, it's also one of these things about sort of teach-to-the-test type thing, is benchmarks.
benchmarks in every domain just are absurdly abused. I mean, this is whether it's the performance benchmarks for computer hardware, whether it's benchmarks for AI performance and so on. I mean, there is
there's just… so, you know, if you say, well, here's the benchmark, and then it's like, let me get my AI to do really, really well on this benchmark. Well, you can start feeding it training data that basically is the benchmark, you can start doing all kinds of things that eventually will sort of squeak it through the benchmark, even though, in more generality, it's really not making it.
And, you know, it's a tricky business, knowing, just as kind of testing of humans, of do people really understand this or that thing? What do you mean by that? And does the multiple-choice test test that? Answer, probably not, because you can just learn how to sort of monkey through the test, so to speak.
But, anyway, so it's a complicated issue, I think.
James asks, if software is open source, who really owns it?
Well, that's a good question. I mean, people… it's a… it's a mess. I mean, there's… I'm reminded of one
open source package.
That, for 30 years.

the main authors of it have wanted us to make use of it for some specific functionality in Wolfram language, and we've wanted to make use of it, but
when it comes to our legal department, and is it okay for us to use this, we want to know, because it requires, sort of, an exception to the, sort of, published license agreement for the software, but the people who actually wrote the software say, yeah, we'd like you to do this. Say, okay, so write us a letter that says we can do it.
Said, well, we can't really do that, because it's really been put
out in the sort of public thing, and we don't really know who can sign that letter to say that it's okay to make an exception to the usual license agreement. So, that's kind of a particularly messy case, where it's kind of like, you don't know… there's nobody out there who can actually agree, because nobody actually owns the thing, and nobody can sort of
you know, it makes… even though it makes no sense, because it was sort of put out… well, it's almost free for people, but there are these exceptions, and it can't be used for this, and it can't be used for that. Well, once there are those exceptions, you better have set it up.
Except they didn't in this case, set it up so that ultimately, somebody is capable of saying, well, actually, you can do this or that.
So… it's… I mean… You know, it…
it's always tricky. There are different, you know, there's sort of a standardized set of license agreements that people use, you know, the GNU public license, and the lesser GNU public license, and Creative Commons licenses, and so on. And some of these, I think, have been sort of fairly carefully crafted. Some of them were originally written as kind of acts of rebellion, in a sense.
And so they… they maybe weren't as carefully crafted in terms of making sure that there are these threads where you can… you can kind of, make changes and things like that. I mean, it kind of reminds me of the constitutions of countries.
Where it's like, okay, we've got this, this constitution, but maybe there's going to be something, you know, and that's how we're going to govern our country, but maybe there's something wrong with the constitution, how do we change it?
I remember looking at this a number of years ago, because I was interested in, kind of, what would happen if you tried to make an AI constitution that's a slightly troubled kind of concept to begin with, but let's imagine you were doing that. It's like, let's say you do it, and that constitution is burnt into the code of all AIs, and the AIs are sort of in control, then how do you deal with that? Because, you know, in 2026, 2027, whatever.
We create the AI Constitution, we burn it into all AIs. The AIs say, this is what… these are the rules, this is how we're going to operate.
And you say, but wait a minute, those rules weren't right, we want to change them.
But the AIs are like, we're in charge, and these are our rules, and how do you change them? So it's a question I was curious, for constitutions of countries, how does… how does change actually happen? And there really were three cases that I found, looking through the constitutions of a hundred or so countries. One case is.
a supermajority of some sort of governmental democratic thing can change the Constitution. That's one model.
Another model is there's some ultimate supreme ruler, often some hereditary monarchy type thing, or something like that, and the supreme leader, supreme ruler, can change it.
And then the third case, which I found the most amusing case, is a lot of, sort of, post-Soviet

constitutions, where basically what it looks like is, to change the constitution, you form a committee to consider the possibility of making a committee that can make a committee to change the constitution and so on, and you realize that it's kind of… the thing is sort of wrapped in an infinite chain of committees. It's kind of like if it was in computation, it would be a computation that never halts or something.
Now, in practice, I'm sure, in situations like that, what really happens is a matter of sort of person-to-person interaction, corruption, whatever you might think of it as, that actually determines how that works. But at least nominally, it's kind of a story of kind of an infinite chain of committees needed to change the Constitution.
But… but so, that's,
that's kind of, you know, how do you do it when there is nobody in charge, so to speak? I've also wondered, when it comes to AIs, for example, right now, kind of AIs tend to have owners. And, in the end, you know, oh, you know, you can't say, oh, my AI did it, it's… well, this is going to be interesting, because it'll be tested, I'm sure. You know, my AI did it, I'm not responsible. Is the manufacturer of the AI responsible? Is the owner of the AI responsible?
you know, if your self-driving car does something very bad, I mean, in current times, it's… there's sort of really only two cases. There's a supervised self-driving car, where you kind of got to be paying attention, or the thing, you know, the little camera that looks at you kind of squeaks at you and says you should pay attention, but fundamentally, you're responsible for paying attention. If something crazy happens.
You're presumably ultimately responsible for that.
The other cases, you know, it's a… it's a robo-taxi of some kind, and you're sitting in the back. And, then I think it has to be the, sort of, the company that's… that's fielding, that… that is… that is making that… that is, sort of, providing that taxi, that has to be responsible. But, the, the thing, but there's this question of, you know, do AIs… will AIs always have to have owners, or could AIs be sort of acting for themselves? And, you know, I did think a number of years ago, for example, with companies, say an AI is owned by a company. The question is, who owns the company?
And in the end, sort of companies have owners, but you could imagine a very bizarre case Where a company, you know, it's an LLC, and it has owned by another LLC, and if you trace through the chain of LLCs, eventually it's just a loop, and it's sort of a free-floating, sort of. bubble of LLCs that are all sort of circularly owning each other. Could that happen? And then what would happen? Who would be responsible if you said, well, you know.
there's… there's, you know, what… who decides something? If the LLC… if the AI does something terrible, and somebody wants to hold somebody liable, and it's this sort of floating bubble of LLCs, how does that really work? And I think in the end, there is… there's always an attachment point, because, for example, if you incorporate a company, there's ultimately, for example, an incorporator, often a, you know, lawyer, law firm, whatever.
else. But in the end, there's some humans involved somewhere, and in the current state of the law and so on, in the end, you'd probably go and pull the thread until you eventually get to a human. Now, similarly, if you have something like a non-profit, you still have a board, or trustees, or something like that for the non-profit, so there's a thread you can pull where the end of it is a human.
Now, you know, in the interesting case of this particular open source software I was mentioning earlier, the… somehow, and I think this was operating under German law, which maybe is less

well developed or differently developed from things that I'm more used to with U.S. law, but where it had sort of been put
in this kind of open-source bubble, where there really weren't any humans who could pull on threads to get anything to happen. And I don't know whether… I don't know if that's really legally correct.
never really pushed it hard enough to know, but that's sort of an interesting question. If something… if something goes horribly wrong… well, for example, one… one very obvious point
is if there's something which is sort of open source software, but it's an egregious violation of some copyright, patent, whatever else. You know, how does that work?
If you've sort of thrown it out into
the open world, and but nevertheless, it is clearly somebody else's intellectual property. How does it work? Well, the answer in practice is that the original intellectual property owner will go after the person who threw it out into the open source world, even if the thing that landed was… is some…
something that says, you know, I'm open, and I'm free, and so on. Well, it wasn't really right. Somebody did something wrong to get it to that point. Now, I mean, the other question is whether, if somebody then picks it up and uses it, as a, you know, oh, it's open, it's free, and I was told it was open, it was free, and I used it, and I put it into my piece of commercial software, and then the original intellectual
property owner comes along and says, wait a minute, you can't do that. You know, you can't make millions of dollars selling this thing that's basically my intellectual property that somebody tried to launder into some open source form and do that. And indeed, I suspect that that would be kind of how the legal side of it would say, no, you know, the fact that somebody tried to launder it through this open source thing really doesn't create a reason why your sort of intellectual property is broken. I mean, there are plenty of cases. I mean, this is certainly something, I don't know, Wikipedia, particularly in its earlier days, was a very common laundering site.
for, you know, take things that were in, kind of, some obviously copyrighted thing, including a bunch of material of ours, I might say, and just sort of put it there, where it kind of is, doesn't really seem like it has an owner, and then make use of it somewhere else.
So that's… it's kind of a… kind of a, a,
You know, a bit of a cheat.
Anyway, let's see…
Jason is asking…
are saying, today there are several prompts that prevent the AI from agreeing with you, and always telling you that your ideas are great, so it becomes more reality-based. Do you think this kind of intellectual friction is essential for AI to become genuinely useful, rather than merely persuasive?
I don't know. I mean, if you deal with people, some people are always very contentious and always disagree with you, some people are always very agreeable and agree with you, and I think, you know.
You know, one can work with both of them, and in some cases, you know, just because they agree with you doesn't mean you should say, oh, that means my idea is right, and just because they disagree with you, that doesn't mean you should assume your idea is wrong.

I think adapting to the personalities of humans is similar to the adapting to the personalities of AIs. I don't really see a, I mean, maybe some people who see AIs flatter them aren't used to being flattered, and maybe that creates a more complicated dynamic.
than it might. Just as I know from, you know, just the world as it is, that there, you know, in, I don't know, some, I don't know, intellectual academic field, there might be people who basically have never had people disagree with them, or never had people sort of point out why what they're doing might not be right.
And so it's… it comes… it's sort of very shocking to them when that happens.
And probably, similarly with,
you know, in a lot of situations, I'm not sure that people are really exposed to lots of people saying, oh, that's a really good idea. I mean, somebody like me ends up with more of that exposure, because I'm generating ideas all the time, and working with people on them, and so, you know, somebody like me is much more used to people saying, that's a stupid idea, or that's a great idea, or whatever, many times a day.
But I'm not sure that that's the typical case, and people who are using AIs, and where the AI is telling them that's a brilliant idea, you know, that may be the first time in the year that they've heard that, and so it probably has more effect than it does if you're, you know, dealing with ideas all day, every day, so to speak.
Let's see…
Elsie is commenting, using AIs and LLMs for anything real, really novel, requires a lot of pushing back against the training and biases. Yes, absolutely. It's, I mean, it is…
sort of an art of some kind to actually get useful stuff from AIs, and I think it's like… like many kinds of things, the…
Sort of… it takes effort to learn how to use the tool effectively, and… but it is confusing, because
It will confidently assert things, even if you're using it very ineffectively.
I mean, one of the things happened to us a few weeks ago is LLM systems connected to proof assistant systems that claim that they're really giving, you know, the LLM is auto-formalizing something and making a precise proof that can be step-by-step verified. Well, we tried doing that on some things, and it said, oh yeah, I've got this proof, here it is.
It's great, it's all verifiable. It was obviously wrong.
And just… just common sense told you it was wrong. The… the sort of the length of the proof for the complexity of the thing it was proving just didn't match up at all.
And the question was, well, what did it do wrong?
And it was very insidious, and I'm not sure we even know, in some cases, what it did wrong. I mean, in some cases, it went through and it was making a proof
And somewhere in the middle of the proof, it said, let's introduce a new axiom. And the axiom was basically tantamount to giving the final result. So it's like, you can't make a proof that's very interesting if you say somewhere in the middle of the proof, let's just assume the result.
But I think in other cases, it was subtly not
sort of defining the thing to be proved in the correct way, and so it was proving something. In fact, I saw it do that a bunch of times, that it… it, I'll give you an example. One thing I've studied a bunch is this Rule 30 cellular automaton. It has a very simple rule, but the behavior you generate is very complicated, and
I've never been able to prove much at all about that behavior. I put up some prizes a few years ago for people to prove certain properties of the pattern you get from Rule 30. Like, for example,

that there are an equal number of black and white cells in the center column, that it's sort of random, at least at the level of having an equal amount of black and white in it.
Well… That's something where you can ask an LLM, go prove that.
and the LLM will chug away, and sometimes it will say, oh yeah, I've made progress, I've done things. You look at what it did, and what it did was completely trivial. It did things like… that amounted to, well, I ran 4 steps
of the cellular automaton, and I found this pattern of bits, and they happen to be an equal amount of black and white, which is completely not the point, because you're interested in, in an infinite sequence that was generated from this infinite pattern, do you have equal amounts of black and white? So it kind of… it kind of cheated. It's kind of like.
I suppose I can imagine somebody who is, well, I can imagine lots of kind of cheat situations where it's like, look, I proved something.
But it wasn't the thing that you were originally asked to prove, but if you don't look too closely, it's like, wow, the AI proved something. It's solved this really difficult, you know, problem, so to speak.
So, it's, I think one has to use common sense about, for example, the AI produces some results. And it's just… it was too easy. It's like, no, that can't be right. Now, how to give people that kind of common sense, how to educate people in kind of the modern AI world, is an interesting challenge. I've thought about it a little bit.
I mean, I think that having, sort of, the intuition for how the world works, what's plausible, what's not.
As important, as is kind of crisping up one's formulation of things, and really giving what is essentially a computational formulation
maybe actually a morphing language code or something, but maybe it's just a more, sort of, formally specified thing than just throw a bunch of words at the AI.
I mean, I… it's very common that if you say to the AI, make
this thing, and you're not very careful about what the thing is, and if there's a cheap way to make the thing that didn't really do the difficult thing that you were imagining doing, then, you know, if you say, oh, I don't know, make
a…
make a thing where the result of the computation is this, this, and this. Okay, and you say it's going to figure out some computation, it's going to generate this sequence of numbers, or whatever else. I'm going to see how the AI brilliantly makes a program that makes the sequence of numbers. And then the AI comes back and says, my program is…
First number is 1, second number is 14, third number is whatever, and the program is just basically writing down the sequence of numbers. And you say, that's a cheat.
And… but if you didn't look at it that carefully, and the AI might have obfuscated the whole thing, and it might be that when you untangle the program that the AI wrote, that it really is just, first one do this, second one do that, third one do this, third one do that, etc. So it really, it quotes, cheated.
But in your definition of just write me a program that does this, well, that's a perfectly valid program. You have to put a lot more constraints on it to make it not be able to cheat, so to speak.
And so that's, again, that's kind of a version of… you have to be… sort of, you have to have more intuition for what the AI might do, and so on, to be able to actually guide it to do something that's really useful to you.
Let's see… Leon asks, what would you think

about when, say, a piece of code, music, digital art is deployed by an ownerless AI and sparks a massive dispute, anything that if a human created it, it would be controversial or offensive, where does the accountability actually live?
That's… Again, I think it's… okay, so…
the AI produces some terrible, shocking, horrible… illegal content.
I'm not sure… okay, so that's an interesting question. What… what… how does that untangle? So… In…
the US, for example, so far as I know, it's, you know, in the sort of pre-speech-ish doctrines, you can kind of say whatever you want.
The thing you say, You better not be, sort of, Presenting it as… Kind of a,
You know, if it's something which is copyrighted, for example, that has certain restrictions. But, you know, the thing, the scenario, I suppose one can imagine is the AI, I can think of all kinds of funky things.
Well… they're very simple things, like the AIs writing some, totally scurrilous thing about some company, product, person, whatever else. And you say, that's defamatory, that's, you know. it's not true, and it's, you know, and it's shocking, and it's, you know, actually, this person is a shape-shifting lizard, and, you know, that's a shocking thing to say. And the AI says that.
And then I suppose the scenario is the AI tweets that, and the tweet becomes viral, and everybody then goes around believing that that person is a shape-shifting lizard.
And then you ask, well, who's kind of responsible for that?
and, you know, that's not really a case where there's It doesn't quite carry the,
the force of having a, you know, it's like all kinds of crazy things get said on social media, and there isn't really, like you said, a shocking thing, and so that's bad. At least not in the US. I mean, there are other countries where
Things are considered to be… it's just illegal to say that.
You know, to make some statement about this or that thing, whether it's about the government, or whether it's about some policy, or whatever else, it just… you just can't say that.
And so what happens if the AI just says that? Well, this is something that's certainly come up for AI companies and so on in those countries, and I think it's sort of a thing that's being steadily attempted to be hashed out.
But let's say that the AI manages to avoid… I mean, there are different issues. One is, is it doing something which is criminal in that country? Another is, is it doing something which is… kind of generates liability, civil liability or something, to… to somebody there? And, you know, those are… those are different cases, and I'm… I have to say my amateur
on-the-fly lawyering is, is, is difficult here. But,
I think, this question of, let's say the thing, sort of.
from its… from its ground-up training data in a way that no human engineer could ever foresee, the thing generates some shocking statement. And that shocking statement becomes widely seen. Then the question is, kind of…
is that who's… who's responsible? Well… If… If a human
said, well, I'm going to look at the 10 things this thing produced, and I'm going to pick the one that I think is going to make the most viral tweet, then obviously that human is responsible.
If it was just agent, you know, the agentic system just spontaneously itself came up with the thing, it's an interesting question.
Let's imagine that you could go back and you found out that the prompt said, make some terrible statement that's absolutely illegal in country X. Let's say the prompt said that.

and then the AI goes and does it.
Then, presumably, in that case, it will be reasonable to say the person who prompted it to do that is responsible for what it did.
If, on the other hand, the person with all the best intentions says, you know, write something about the history of this and that and the other, and the thing as a result of that generates something incredibly shocking, who's responsible in that case?
I think it's hard to assign, kind of, responsibility unless there is some kind of chain where somebody let it out.
and knew they were letting it out. I mean, I suppose it's the… it's the issue of, did you do it on purpose, or was it an… was it an accident?
In some sense, what comes out of the AI, at some level.
might be thought about as being like an accident. So, let's peel this back a little bit in a more foundational way to think about, you know, when are you really responsible? You take certain actions, and certain things happen in the world.
And the question is, are you responsible for what happened? In other words, if you, sort of, put… a…
I don't know, you put a banana peel somewhere, and that's probably a bad example, because you don't put banana peels anywhere unless you want people to slide on them, so to speak. But let's say you did something sort of
with the best intentions, you put
I don't know, you parked your car somewhere that was a perfectly reasonable place to park it, and then that blocked an escape route for somebody who needed to rush away from some fire, or something like this. And then somebody says, that's probably a bad example, but, you know, in other words, you could not have foreseen that the thing you did would cause that harm.
And so then the question is, and I think the law has a lot of, kind of, provisions for this, is can you foresee what would happen?
If you can foresee that what would happen, then you're sort of willfully doing something. You're doing it, you know, because you knew it was going to happen, as opposed to you did it, and there's no way you knew that that was going to happen. It was just one of the things that happened as a result of things you couldn't foresee.
Well, my own efforts in studying, kind of, systems and computational systems identify this phenomenon of computational irreducibility. And what computational irreducibility is really about is you set something up in a certain way, can you foresee what will happen as a result of that setup?
Well, it could be the case that you can sort of think ahead, and you can, given that setup, you can absolutely predict this is what's going to happen. Or it could be that it's computationally irreducible, and that to know what happens, you just have to run each step and see what happens.
So, in the case where there is computational irreducibility, one could argue that it's hard to hold the originator liable. In other words, you could say, I set up these rules, but I couldn't know what they would do.
And… and I couldn't have known what they would do.
So it's something that is really just an accident, or something which, you know, in other settings will be thought of as sort of an act of God. It's something that comes in from… from… it is not something that is a thing induced by… by human effort, so to speak.
Even though you might have set up those underlying initial rules, but you couldn't have foreseen what would happen. Now, you could say, but the very fact that you can't foresee what will

happen means you shouldn't have done it this way. That is, if you say, I'm going to, kind of build this
piece of a building, and I set it up in a computationally irreducible way so I can't tell what would happen. People would say, well, that's not good enough. You have to be able to tell that it won't fall down.
In order to be sort of validly signing off on these engineering plans for this building. It's not good enough to just say, well, I set it up this way and I don't know what's going to happen.
But there are plenty of other cases where you set it up in some way, and it's… it's not…
you're not guaranteeing what will happen. It's not the nature of the thing you're doing to be able to make that guarantee. So I think the answer to this question of, you know, when are you responsible, is if there's a sort of
think if what you did was what you had to do to get any of the results that you wanted to get in order to… even to run your AI, that you needed to, sort of, put yourself open to a bunch of computational irreducibility.
then, sort of, just to do what you were trying to do, which was a reasonable thing, let's say, you opened yourself up to computational irreducibility, but there was computational irreducibility, and so you couldn't have foreseen what that would do. Now, if you insist, well, you have to check the results.
then obviously you're on the hook for checking the results. But if it's just, like, it was just run, and there was computational irreducibility, and you sort of couldn't have known what would happen, then I think it would be fair to say you're sort of off the hook. That's then…
See, in a sense, it's related to human responsibility as well.
If you say about a human, you say, why did you do that terrible thing?
And the person says, well, it was just something in me that made me do that terrible thing. Then you would say, well, you're responsible. There's nothing… but then the person could say, well, I did that terrible thing because
you know, I, pushed that button because somebody was pushing my hand to push that button.
And then you say, well, that, you know, it wasn't me that was responsible, it was the person who pushed me who was responsible. And that sort of generalizes a bit more to say, you know, this terrible thing happened to me, and therefore, sort of the echoes of that thing that happened to me caused me to do this other terrible thing in the future. So, in a sense, it's not fundamentally me that's responsible.
It's fundamentally this external thing that happened X number of years ago, which was not me, but was imposed on me, that caused me now to do this other thing.
And then there's the question of, sort of, does that absolve you of certain responsibility, that there was this thing that imposed itself on you that then had this cause for you?
So the same kind of thing will happen for the AIs, that is, to what extent is the thing that is coming out the result of, sort of.
internal grinding inside the AI, sort of irreducible, computationally irreducible grinding inside the AI, and to what extent is it somebody kind of tipping the scales, putting their finger on the scales, you know, late in the process?
So this came up.
I did some testimony, actually, for the US Senate back in 2019, where some of these kinds of things came up about, kind of, the ranking of content in social media and search engines and things like that. And there's sort of a question of if you can see the code that did the ranking, does that then allow you to know how the thing will work?

Well, in some cases, you could see the code, and if the code has an if statement in it that says, if it's John Smith, you know, hide it, and if it's Fred Jones, put it at the top, and it's right there, and you as a human can see that in the code, then yes.
you can tell that something sort of scarless is going on. But if what's in the code is just some ground-up neural net
Type thing. You can't…
tell whether something scarless is going on. Now, again, if you looked back at the cause of the thing, if you found that reinforcement learning had been done, that somebody had made a neural net, and then they'd run a reinforcement learning loop that had tried to make sure that Fred Jones goes to the top type thing, then indeed, again, one could say, yes, that's a place where, sort of. finger was put on the scales, you could see what happened. It's something imposed from the outside. But if it's something which just sort of… the whole AI was sort of created from all of its disparate sources and so on, it just learned from its natural life, in a sense.
And nothing was really… no… no fingers were put on scales type thing, then it's, it's much harder to be able to sort of attribute blame for what happened in the system.
Let's see…
George is asking, what's behind the fact that current AI systems can produce completely different levels of reasoning and accuracy when asked the same questions in different languages?
Well, I think the point is that one might have thought that
there was some sort of intrinsic language-independent representation of meaning out there that the AIs are plugging into.
actually, we don't really know if there's some intrinsic, universal, language-independent kind of framework of meaning that can exist. Actually, I've been interested in that question for several decades, and actually, we have a project that's been running for quite a few years now.
to really nail that down, and to try and sort of build a layer, sort of a language-independent, sort of symbolic representation of meaning that doesn't depend on the words in English versus the words in French, and so on. But it is not self-evident, and it hasn't been philosophically, that there is
a meaning that there is something which is sort of pure disembodied meaning, independent of language. And certainly the way LLMs work
they are really based on doing things where you're… you're looking for, sort of, patterns represented in language. Their version
of how the world works is the way the world is described in language, or if they're dealing with images, in images, or whatever else. They are dealing with, sort of, representing the world, they're sort of pulling back
from… they have an internal representation of the world that is derived directly from, sort of, representations of the world in things like natural language. Now, you can ask the question, did they somehow form this
kind of language-independent meaning structure inside the LLM. Nobody really knows that. And it's…
in some sense, some of that is probably being formed, but it's not really clear, and in some sense, it's just, well, I can reason in any language. I mean, if you think, I mean, I'm not fluent in any… in any human language other than English.
So I'm not in a position to think in other languages. I mean, I do notice that, I know people, when… it's sort of an interesting phenomenon that if somebody asks you a question, okay, so I can just about manage in French.

So somebody asked me a question, and…
I might just take that, translate that question, and translate it in my mind into English.
then know the answer, translate it back into French, and say the answer. But I know that in my hopelessly unfluent French.
you know, I can get to the point where I can be just answering the question without at least consciously translating the question into English to be able to know the answer.
And so, somehow, in my brain and other people's brains, there's a way of thinking, sort of direct… whatever the process of thinking is, it can happen in other languages. And that's presumably what's happening in LLMs, that the thinking is very language-based, as it is, I think, in humans, and the thinking can happen in different languages, and sort of the training for thinking is a bit different in different languages.
languages. There's been often, you know, English is the one where there's by far the most material that's available for training, and it could very well be the case that the material that's available in French has certain
prejudices, for example, that are different from the ones that, if it was sort of born English, it could get. Now, the way that, sort of the
the attempt to turn the training data into some sort of internal feature representation that is closer to a pure meaning representation. That, presumably… it's the fact that it's gone a certain distance towards that that allows it to take a question that was asked
in, you know, Swahili or something, and use what it learnt in English to be able to answer that question. That the, sort of, the pattern of what was going on internally was
was somewhat universal, but this is a… it's a complicated area, and there's much that is not yet understood about these kinds of questions, and about the extent to which there is kind of a language-independent sort of core of meaning that LLMs
Either have implicitly.
Or you could extract explicitly. I don't think they have it explicitly. I don't think there's a mechanism where you can point to a piece of the LLM and say that's how it knows that logic works. I think it's all kind of very much these kind of lumps of irreducibility that sort of happen to fit together in such a way that out of it comes the working of logic.
And that somehow that… that you kind of get to that same place, starting from different languages and so on, but maybe in slightly different ways.
Let's see…
Well, here's a much more practical question from LC. What's your view on how to go about AI or compute infrastructure as a tech company, whether and when to buy it from big guys.
CapEx versus OPEX and so on, capital expenditure versus operational expenditure. In other words, are you buying something once? Like, I'm gonna buy a bunch of servers, or are you sort of just renting time on somebody else's servers on an ongoing, you know, month-by-month or whatever basis?
Well, it's an interesting question. I mean, there are a lot of different kind of curves that are moving in different directions. That is, it's like, well, you know, models are getting more efficient, so it's getting cheaper to run things.
I think there are a number of thresholds.
One question is, for the thing you want to do, Is it reasonably doable
with a model that could even run directly on your computer? Is it runnable on something which is a not terribly specialized server, or is it only runnable on something with a huge amount of memory and lots of GPUs?

So I think it depends on the task, and it depends on the time, and as models have become more efficient, and, you know, quantization of models, or compression of models, and so on has become more efficient, it's moving towards models of the kind you want can be run on smaller and smaller computers.
Now, it could always be the case that there's always a bigger model.
And you could say, if I always want the best model, I have to go and get it on this big, expensive computer. But depending on your use case, eventually, you can get to the point where, for that particular thing, you know, summarizing emails or something, or, you know, doing,
I don't know, spell checking, or doing, or ranking some, kind of basic search results, or doing, you know, this or that. That these things come over the horizon, so to speak, to be things that are doable on a fairly, low-level or local computer.
So I fully expect that Okay, so, so, you know, how is this going to evolve?
the thing that LLMs definitely do well is provide, kind of, linguistic user interfaces.
But… what…
They increasingly… the increasing way that they get used is as those linguistic user interfaces to underlying computational tools, like our technology, for example, or like technology that actually, you know, moves files around you on your computer or whatever else.
They're increasingly… they're that user interface layer.
That provides a chat user interface that's very, sort of, human-friendly and so on, but the real meat of the thing is happening in the tools behind the scenes.
If that's the case, that linguistic user interface layer will surely become something that is routinely run locally on your computer, your phone, whatever else.
I think that linguistic interface layer will not need to go to all the fancy, expensive places.
Now, if you're doing sort of things that involve, let's say, coding, where that so far is still making use of more complicated chains, I mean, what's happened there is that what was originally lay down the next word in a document is now much more lay down the next task. So instead of next word, next token prediction, it's next task prediction.
And that, right now, is…
requiring larger models and more, kind of, computational effort and so on. I mean, the issue with models
tends to be that they can be very big, and if you want them to run efficiently, they have to be in memory. And so that means you have to have machines with a very huge amount of memory, and that's something that your typical laptop and so on is not going to have.
But, you know, models that provide the linguistic user interface layer will get smaller and smaller, I think. I mean, there's a lower limit, which is sort of determined by, in some sense, the fundamental information content of natural language, but I think it's not that big.
And I think it will be no problem to fit that on laptops and phones and so on, if the main thing is just that interface layout.
So…
you know, in terms of the actual operation of these things in companies, there are a bunch of different issues. I mean, one is, where are you prepared to let your, you know, trade secret stuff or your user data that is protected by privacy rules and things like this, where are you prepared to let that stuff go?
And…

Sometimes, you know, there are… it's either… it's got to be in our… in our… on our premises, in our computers is one level. It's in, kind of, the… the trusted generic cloud providers, that's another level.
It's in kind of these somewhat new companies that are sort of really AI companies.
Or it's in some, you know, latest startup that just got their venture capitalists to give them enough money to be able to subsidize running, you know, AI models for less money than other people, or something. So there's different levels of trust that you might want to, sort of apply to those different kinds of companies. And I tend to think that there are also cases where what you are getting from a generic cloud
provider is sort of just a container, and in that container will be enabled one of these AI models, but really, all that the sort of the cloud provider sees is the outside of the container. They don't really get to see what's happening inside the container. That's one model. Another model is where sort of the bits are flowing through their systems.
in a way that they could easily be getting or choose to get telemetry from those systems. So there's sort of different levels of
Of, you know, where the data is going, and how much trust you put in the providers, and what's happening, and even what the agreements say about what's, you know, what's going to happen to the data, and so on.
I tend to think, that… there are…
You know, in terms of how much it should cost to do all these things.
You know, it's a complicated story right now about the cost of AI stuff, because there's many different dynamics. One dynamic is companies rushing in to try and get users by heavily subsidizing whatever sort of AI
stuff is going on. That's one case. Another case is companies worrying about what they look like to investors, and particularly to the public markets, and saying, wait a minute, we've got to be charging enough that we actually make money from all of this stuff, so the prices go up.
And the prices can go up to be generating large profits, versus, you know, what can happen in the other case, where you're sort of trying to buy market share and just get users, and where your profit is definitively and purposefully negative.
So, you know, there's a question of sort of what will happen to those prices, because there are two different forces going on there. What you see happening a lot in the AI world right now is endless hosting providers. And, you know, often, you know, companies will be created with the best of intellectual objectives of some very elaborate sort of story about the amazing intellectual things that are going to happen.
They get a bunch of venture capital, and then, in the end, they turn into hosting companies, where that venture capital is being burnt
essentially attempting to buy market share by selling the kind of AI inferencing at sort of below cost or whatever else. That's another complicated dynamic.
But, you know, right now, there's a… you know, it's very much in the… the… people are saying we're raising, you know.
huge amounts of money to build data centers. That's a complicated story, because there's a lot of circular deals involved in that kind of thing, and it's not clear that that is really a matter of we really want to build the data centers, versus we want to have so much money flowing through the system that the things that sort of are connected to the money flowing through will make money, and so on. It's a complicated kind of
Set of things that are happening there.

But… You know, and then there's the question of if… if…
AIs get more efficient, what happens to all the capacity of these data centers? If, you know, it,
So… my… Fundamental answer right now is that if you can get, sort of.
if the things you want to get from AIs are simple enough that you can run them on reasonable cost hardware that you might already have, then why not do that? It's,
You know, I… we haven't yet
invested big time in local AI hardware. We've invested a small amount in that. I mean, for example, for something like Wolf Alpha, we never used outside hosting providers. We've always had our own data centers and so on. That's partly because when we started
well, from our first 17 years ago now, we, sort of the landscape of cloud providers was a bit different. At times, we've looked at, you know, what would it cost to run that kind of thing on cloud providers, and the answer is it's more expensive there.
And sometimes people at these cloud providers have said, how can that possibly be? We have so many economies of scale. But the truth is, you know, sort of, so do we, in a sense, but they're different economies of scale. And once you have a service where, you know, it's being used at a, you know, on average a constant rate.
You might as well just buy the hardware. It's cheaper than renting it from other people.
Anyway, the, there's much more to say there, but I have to go to my day job in just a second here.
Okay, X7 is asking, for privacy purposes, would it be possible to do the embedding locally and send, via an API, a vector of numbers instead of the text?
That's…
an interesting thought, but I don't think that really works, because the embedding is not cryptographically secure. In other words, it's not set up so that… I mean, this is the… okay, so the issue is, if you say.
You know, normally you'd just be sending plain text.
over the API connection to the model, or whatever. And then you say, well, somebody could wiretap that and just see everything I'm asking, you know, asking the model. Say, well, don't do that.
Do the first level of embedding so that you're actually getting just sort of feature vectors of your text, and, you know, token numbers and things like this, not your actual text.
And so then, what you'd see on the wire is not something that's obviously a piece of English text, you'd see gobbledygook.
The problem is that that process of going from the text to the gobbledygook
is not cryptographically secure. It's something where you could perfectly well know what that process was, and invert it, and say, given the scobbledygoop, that's what, that's what my original text was. In fact, that's exactly what an autoencoder does. It's trying to take… go from a message back to a message, going through some set of features in the middle. So it has both sides of the… from the… from the message to the features, and from the features to the message. So I don't think that quite works.
And I think the, in fact, when I was, doing that testimony for the U.S. Senate back in 2019, I was, kind of…
had thought of this way of, kind of, having different, kind of, final ranking providers for content
And I was thinking of basically being able to have embeddings that get sent to those ranking providers and so on, and I had, in fact, in the sort of night before I was doing this testimony, I'd kind of checked with people I knew at the large social media companies and so on.

To, to find out, would this… would that scheme actually work?
the response was typically, yeah, it would work, but, you know, and some of them were saying, actually, it would be good if that was regulatory required, because then we'd have to clean up our code base. But yes, it would work, but the code is such a mess, it wouldn't be easy to implement.
But, so that's… that's… but I think the… the only way of sort of sending something that is encrypted is this idea of fully homomorphic encryption, where sort of everything that is seen on the outside is encrypted, and somehow computations can be done on the internal encrypted thing.
And… and still be done correctly.
doing that for machine learning, one doesn't know how to do that at this time. And in fact, the idea of fully homomorphic encryption has become a bit diluted, and what people often mean by that now is not, sort of, the full story of what is thought to be cryptographically secured.
I think
Yeah, it's,
Elsie is commenting, it's damned if you do, damned if you don't, in terms of letting big models have your best data. The risk of falling behind is getting more existential versus the risk of a leak of a proprietary edge.
I think it's a question of, you know, if you're doing, sort of, the most sophisticated analysis of data, you know, do you…
let… I mean, the agreements for most of these models do not allow them, if you don't check the box, so to speak, to train on the data that you send them, but there's a question of, you know, do you really trust that, and etc, etc, etc. And
Yes, I think it is a… it is a complicated thing of, you know, what, what you keep internally versus what you allow to be potentially seen. And I know at our company, we have a bunch of policies about this that we've, that we've worked out.
And a bunch of, kind of, technical, kind of, constraints that let some things happen and not other things happen, and so on.
Anyway, time for me to go back to my day job, and, thanks for a bunch of interesting questions, and
for getting me to do a certain amount of amateur legal thinking and so on here.
And, hope you all found that interesting.
Well, thanks for joining me, and see you another time. Bye for now.