All right, hello everyone, welcome to another episode of Q&A about business, innovation, and Managing Life.
And, talking of life, I'm away from my natural habitat right now, and I'm in a slightly remote location where
internet connectivity is not as good as it might be. So, if things freeze or fail, I apologize in advance, and … we'll decide if we can actually continue or not.
But so far, it seems to be doing okay, for at least a hot 30 seconds or something. All right, let's get into questions people have.
… Let's see…
Jace is asking, how were your early years in academia, attending university at a younger age than the norm? Would you say it was unique?
Well, I don't think it's unique. I think the… I mean, I… I…
I left high school when I was 16, went to college at Oxford, left there when I was, like, 17, and, went to graduate school at Caltech, and got my PhD when I just turned 20, and, then…
Was a faculty member after that.
… for a while, until I stopped doing that and decided to start being a tech CEO instead.
But, so I was…
definitely younger than the average. I would say in all of those steps, I was quite a lot younger than the average. I didn't notice that terribly much. I was just doing what I was doing, which was mostly concentrating on the kind of physics that I was doing, and
particle physics and cosmology and so on, and I was just, oh dear, that's frozen, I thought that was gonna happen.
Let's try that again. Okay, maybe that unfroze. Who knows? … I couldn't find out, …
what the distribution of weird expressions that I make is. In any case, the, …
the… I was saying that the, … … …
I… so I ended up much of the time being considerably younger than other folks who were in the same kind of academic situation that I was in. Again, I didn't really notice that much. I think the…
would I have had a…
better experience if I'd been older? I don't think so, particularly. Oh my gosh, this is frozen again. I think maybe we'll just do audio for a little while here.
… I think the, … The thing, … … Let's see, the…
there are some effects of going through the system young. For example, one effect is that now that one goes 45 years later.
I'm still, I'm happy to say, sort of alive and kicking and energetically doing things, but people who are my contemporaries in that sort of early stage in academia are now very much at the end of their careers.
And that's, it's kind of one of those things, I don't know, it's perhaps a… a, I don't know what it… what kind of a thing it is, you know, may you write many obituaries type thing. Well, that will happen.
If the people who were your, sort of, contemporaries, end up being much older than you, at least demographically, that has a likelihood of happening.
I think… my…
own, sort of, experience in, do I recommend other people to get through things as young as possible? If you have something, a place you're trying to get at the end of the tunnel, so to speak.

then it's a win to get through those things as quickly as you can. If you're doing those things and you don't really know where you're going at the end of it, then it's typically a lose, because you get to the end of it, and you're, you know, 20 years old, and you have your PhD, and then you're like, well, now I've really got to decide what I'm doing.
And if you haven't decided that already, you can't get… you don't get to hide in school for another, you know, 6 or 7 years figuring it out. So it's a much more exposed position. So if you very much know where you want to get to, it's a thing that can potentially make sense. I think in my own case, it was just as well that I got through that whole system very young, because
I might very well have lost interest and generally, become frustrated with the whole thing if it had taken many more years. And as it was, I, was still feeling very energetic by the time I was sort of finished with school. I hadn't,
gone through the whole process of grinding through many, many years of getting a PhD, and then maybe I was a junior professor or something, grinding through many, many years of doing that.
And then pretty soon you're 40 years old, and it's like, I'm finally through all my education. And then it's like, but I'm totally tired out.
So that's a good thing if one gets through the system young, not to have that, that, that happen.
I'm gonna try switching video on again.
And… oh, there we go. Alright, let me see… Anna asks.
Who's going to innovate faster, small startups or big companies?
Well, I think the data is very much small startups. But it's a complicated world, because that hasn't always been true. I mean, at various times in the past, you know, there was a period of time in the, you know, late 1960s, 1970s.
when the U.S. government was the big place where lots of innovation was happening, and lots of national labs and so on were doing lots of kinds of things. NASA was doing lots of kinds of things.
That was a, that was a time when, sort of the big innovation was happening in the big government, so to speak. I think things changed.
And then lots of innovation started to happen in increasingly small companies. It's much easier to innovate in a small company… well, why is it easier to innovate in a small company?
Because you're not… you don't have a bunch of systems already set up that are all built to optimize doing the thing you've already been doing.
if…
if you have those systems, there's a great tendency for a company to just say, let's keep doing what we've been doing, that works, and we have systems for doing that, rather than let's cast out in some weird new direction that might not work. There's a considerable force preventing that happening.
And often, companies' own internal bureaucracy will drag down any kind of innovative direction that's being taken. There'll be sort of internal review meetings, and people won't understand what the point is, and people will say, why are we doing this? A line of business that's smaller than, well, depending on the size of the company, $100 million, billion dollars, whatever.
A line of business that's smaller than that.
isn't one we can deal with, and they might be right.
as a… if there's a line of business that's going to start as a $10 million a year business, and all of your other business units are a billion dollars a year, there's all kinds of systems that have been built for billion dollar a year business units. Apply them to a $10 million a year business unit,

and you'll drown the thing in bureaucracy and systems and structure and so on, and it just will never get off the ground.
Now, you know, it has to be said that in the world of small companies, the sort of the big thing around small companies is investors and venture capital and so on, and when that gets very institutionalized, it too can have this kind of constraining effect.
It isn't a constraining effect in the same sense that it's like, oh, we've got all these checks and balances because we're expecting it to be a billion-dollar business unit. Rather, there are things like, we have this theory about how the world works, and everybody should follow that theory, and if you're doing something different from that theory, no, no, no, that's a bad idea.
And that's not a pro-innovation kind of situation.
I think that sort of having the freedom to be innovative is a critical thing, and in principle, that freedom could exist in a large company. In practice, it tends not to, and in principle, that should exist in small companies, but sometimes it doesn't, because of the constraints of investors.
And I think in, you know, in our own company, for example, doing innovative things is something that we have sort of got experience with the fact that that's a thing we can expect to have happen. Let's just froze again.
… And, …
Once you have, kind of, the expectation and experience that you're going to be doing innovative things, that helps quite a bit.
But I know in my own company that it's pretty common for us to have to start, sort of, a special projects group
that is going to try and do something innovative, and that follows, kind of, different rules from the main part of the company. It's often a challenging thing, and you often get a lot of, kind of, corporate immune system kind of attack.
When you try and build a business unit that works differently than other business units. You might have a more entrepreneurial business unit, where people are like, they're not following the rules, they're not, you know, going through all these review meetings. And it's like, well, if they did, you would have already killed what they were doing.
So, it is… it takes, you know, some sort of management force to be able to have
sort of innovative stuff happen within companies of, well, particularly a company which has… already has kind of structure built into it. I think
The, on the other hand, it can be easier to innovate in a situation where you have the resources, and you're not always having to kind of convince somebody else about every step in your innovation.
Because as soon as you're convincing somebody else, the chance of it being really innovative is much lower. In a one-person, small group.
can have some crazy idea and pursue it, and it'll be right. As soon as you have a bigger pool of people you have to convince, you're going to pull it down to something which has less and less innovation, I think.
Now, I would… I would say the…
The, … it's sort of a complicated thing, because if you give an innovation effort too many resources, you also can kill it.
when people are like… I think it requires a certain… well, it requires… I don't know whether it's quite a hunger, a push to actually, let's make this work, let's, you know, we're… it's an adventure type thing. Let's… let's see if this can work.

As soon as it's like, there's a ton of money riding on this, it's a big effort, it's, you know, it's really important, and so on, people have a habit of freezing under those circumstances.
Plus, if there's a lot of money flowing in that direction, there's much less tendency to say, hey, let's just try this thing. It's not gonna… it's not gonna be too much of a risk to try this thing. It's more like we're diverting many millions of dollars into this thing.
you know, are we sure it's right? Are we sure we are not making a mistake here? So sometimes, the higher resources, even though you might think higher resources, obviously better, it's not obviously better. Quite often, it makes things worse.
I've seen that a lot when people have tried to start research organizations, from within companies,
They, they'll say, of course it'll be better to have this big research organization with lots of, plush, you know, facilities and so on, but it actually isn't.
I think often companies and R&D organizations do the best when they're slightly scrappy and so on. I mean, I remember noticing years and years ago
beginning of the 1990s, I moved our company from kind of scrappy office space to much more lovely office space. And somehow the energy level went down. The scrappy office space, which was sort of a converted apartment building and so on.
Was everybody, it was all a mess, and there were cables lying around on the floor, and all kinds of things like this.
But somehow there was a level of energy, and there was a bit of noise all the time. Move into a lovely office space, and everything was very quiet. It's very, very, sort of, beautiful, but the energy level was lower. And I think that sort of reflected in, …
In the idea that sometimes you need, sort of, enough resources to be able to get things done, but not so many resources that it's all sort of quiet and relaxed, so to speak.
It's a tricky thing, managing R&D in a way that can be effective. You know, I think it's something we've worked on a lot over the years at our company, and now with the Wolfram Institute, I get to try and do it in a slightly different way, because with a company.
there's always… we're making products. You know, it's a thing where, ultimately, we're doing engineering, we're building things, it may take a while, we may have a bunch of problems to solve in the engineering.
But ultimately, it, it's a thing that, we can…
expect… you know, there's a… there's sort of a linear path to get to the end, whereas in research, pure research, that's not the case. It's not obvious what the deliverable is at the end, it's more complicated to know that, and it's also, you're going off in a direction
And that direction, it might be a bottomless pit. You might never be able to get to the end, so to speak. And so it requires sort of a different management approach than one is used to in building technology and so on.
Let's see… …
There's a question here.
from KD.
Actually, I'm gonna have a… yeah, okay, there's a question here. If the young Leonard Euler applied for a job at your company, would he be successful in the interviews? Would he do well on the job as a prolific math person?
well.
I think that, my guess is that Leonard Euler would be very successful and prolific at our company. I think, assuming that

he wasn't, and I think Leonard Euler himself, from what I know of his history, was a somewhat flexible individual who worked on many different things in a variety of different environments. And I think that my guess is that, in…
And I would hope that in our company, he would get enthused about, kind of.
building not formulas to write in papers, but things implemented in code, and will be off and running and being prolific, you know, 300 years, 250 years after… after his time, so to speak. I… I don't think,
I don't think, … well, I mean, if you look at… for our company.
People who've been, sort of, academics, and where the main value proposition is write papers. That isn't a good fit for our company, because our company isn't in the business of writing papers. Our company is in the business of making software that lots of people can use, which is a different problem than writing a paper about
Even writing a paper about writing software is different from writing software. Something that, you know, often in academic computer science, people are writing papers about writing software. And the kind of constraints and objectives there are very different from actually making software that will work as well as possible. So it's, for us, it's kind of like that there's a certain type of person who is kind of into, let's build the software that can work great.
As opposed to, kind of, the, let's, let's kind of, work for the gallery, so to speak.
And build something that we can write things about.
And, also, you know, what will be considered a very fine academic paper about an algorithm may be hopeless in terms of implementation, because in the fine academic paper, you may get the point of the algorithm across.
But, oh, you're not talking about those details over there about how this or that works. You're, … But those details are what's really important in actually having the algorithm be a thing that can be run and people can use. And very often, you know, if we say, oh, somebody wrote a paper about this, great, we can use some idea from their paper.
Well, maybe we can use the idea, but usually the vast majority of the actual algorithm in detail, we have to construct for ourselves. It's not in the paper.
It's… people didn't, sort of, build it as a production thing and deal with all those corner cases. That's something that we have to add.
There was another question, Jeannie.
Did you ever skip class for something more interesting? The British school system is a little different than the American school system, and … I…
…
well, different in many ways. By the time I was in college, you didn't have to show up to any classes at all. It was just a, take the exam at the end of the year and be happy type thing, and there was no show up at classes, and I didn't.
and I… the… when I was in high school, the, … …
I was, again, a bit of a different setting, because I was at a boarding school, a place called Eaton, a fairly well-known school in England, at least. Been going since, I think, the 1450s, if I remember correctly.
And, it was kind of… there wasn't really any place to go other than, sort of, go to class, at least when I was there. In the last year or so I was there, I'd kind of…
more or less finished the classes I was really supposed to be taking, and I did, not unusually go off and take the train to Oxford and go to physics seminars there. So I suppose in that sense, I, …

I, quote, skipped class, although it wasn't like I was supposed to show up to something and didn't. It was, … I have to say, I'm… I'm curious how that actually worked. I guess I had… I… because I disappeared for a whole day, so I don't really know how that, … how that would work. … the, … … Let's see, I think we've… our audio is frozen, but I'll try again.
…
Let's see… I have a bunch of questions here.
…
…
Hmm.
Well, there's a question here. There are hundreds of AI startups. How can I judge if one is actually trustworthy and going to provide better quality answers versus others? I think most you know, the fact is, with the technology of AI, there's a certain threshold that's been passed with LLMs, and it is what it is, and gradually they're getting cheaper, faster, and so on, and a little bit better.
but mostly it's sort of the harnessing of the LLMs that's better. And it's kind of like, if you were using a horse as a way to do… to, get from here to there, or do things, you have a horse. Yes, you can evolve the horse a bit. You can breed the horse.
well, but the thing that's going to give you the most value is finding ways to harness the horse better. What is the horse supposed to do, and so on. And I think that's kind of the situation we're in with LLMs. That, yes, you can breed slightly better LLMs, but a large part of the real, sort of, great value to be generated right now is how do you harness the LLM
I see LLMs as primarily
valuable as linguistic user interfaces to things. Whenever there's a sort of use case that involves the LLM as a linguistic user interface to something, that's going to be a good use case. When it's a, like, I'm going to get the final finished goods.
of some precise thing from the LLM, that's typically going to be a lose. I mean, a thing that is important in, kind of, the direction of LLMs, it's kind of… LLMs are doing things a bit like humans can do, with all the benefits and problems of that.
It's, …
I think, if you ask the LLM to kind of, in its own mind, do a bunch of computation, it's not going to be able to do that.
And that's where you should use our technology, where we've been able to build this kind of, in Wolfram language, this sort of big formalization of the world in computational terms that plugs in very nicely to LLMs, but it's not an LLM as such.
So, when it says… when you say, provide better answers, I would say the companies that use our tech stack and so on to… as tools for LLMs to do things, that's a good bet.
The companies that say, we're gonna build a better LLM, we're gonna remove all hallucinations. It's like, that's… that's just not a thing. It doesn't make sense. It's not even a… it's not… even at a sort of definitional level, it's not clear what that means. It's like, this LLM will tell no lies. What happens if you ask it to write a piece of fiction? What's it supposed to do?
That wouldn't even make sense. There's no way to do it. You can… if you have an LLM that's using computation, the computation can effectively tell no lies, it can just compute in some precise way, but the LLM is not going to have that option.
So, I, I think in, in, …
I don't know, you know, it changes month to month what startups say they're doing.

And, you know, there's been a burst of interest in, for example, various kinds of acceleration of LLMs and hardware and so on. There's been people using… I mean, at the beginning, there were lots of things with prompt engineering, and then there are things with, you know, going agentic and having more connectivity to agents and so on, and then, you know, marketplaces of agents. And these days, MCP is a big buzzword. That's kind of a sort of, kind of standardized way of connecting tools to LLMs and so on.
And, that's a, that's certainly useful to us, in the sense that we can provide our computational tools, through, through MCP,
to LLMs. I would say it's not exactly as standardized as you might hope it would one day be, with only, just a lot of companies saying, we're doing MCP stuff.
But, you know, there aren't… there's… the kind of what's there is a very rough framework. And, even the details of how things work with remote MCP servers and so on, and authentication, all those kinds of things is not properly worked through yet.
Let's see, there's a question here from Caleb about how would you think about regulation around AI?
I think it's difficult. I think, and I think there's many…
it's like, what… what do you get to regulate? You know, do you… for example, here's a… here's a question.
Should AIs have freedom of thought?
Should an AI be able to think about anything it wants to think about? Or should that be somehow regulated?
With humans, humans, in a first approximation, have freedom of thought. What gets regulated is what the humans do in the world.
There isn't a kind of, there isn't a… you're not allowed to think that, at least not, not in,
and it's not an implementable thing at this point. It's more like, well, you can't say that, or you can't do that, but those are sort of actuation-type things, rather than internal sort of thoughts. So with AI, I think it is, you know, people are a little bit imagine that they can regulate the thinking of AIs. That, I think, is sort of a doomed… a largely doomed idea. I think as soon as you regulate the thinking of AIs.
you won't be able to get AIs to do anything terribly interesting. I think there's sort of a trade-off, it's a consequence of the whole computational irreducibility idea, a trade-off between letting a computational system like an AI sort of have its head and do interesting things.
Versus constraining it to always have to follow a particular, kind of, a particular determined, set of constraints, and so on.
So… Let's assume that, AIs have freedom of thought.
Then the question is, how do we affect what they can actuate?
What data they can get access to. What, you know, can they read every book? Can they, collect. you know, all sorts of private data and so on. Then, what do we allow them to do in the world? Can they drive cars around? Can they, launch weapons and weapon systems? Can they, you know, determine who gets a loan and who doesn't?
These are things which are kind of actuation layer things. And there's sort of a question, I suppose, of what things do you want a human to sign off on, and what do you not? Somebody isn't realistic, because it's going to happen in a millisecond, and there's no choice but to have an automated system do it.
But a thing that's very well developed is, you know, if you're an engineer building a bridge.

then you can use whatever software and tools and techniques and so on you want to, but in the end, you're the one who's signing the documents saying, I believe in this bridge, and you're then responsible for it.
Now, you know, the question is, do you unhook that, and do you say, my AI is going to do it for me? And under what circumstances do you do that? And yes, there are cases, whether it's driving cars, or whether it's doing various kinds of bidding-type things.
Where the speed of things happening is sufficiently great that a human couldn't be in the loop. So, it's sort of, you have to say, oh, you know, are we going to let the AI do this or not, or are we going to have the AI say, you know, pop up a, you know, an alert for some human saying, press this button to commit to doing this, or not?
I think that's the place where one can sort of say, well, there has to be a human in the loop for this particular kind of thing, or not.
I think the idea… so, you know, there are many other tricky things, like… like AIs, increasingly, you know, humans are the actuation layer. The AI is talking to humans and is convincing the humans, go do this terrible thing.
what is the line of responsibility in that case? You know, is the maker of the AI responsible for the fact that this AI succeeded in convincing this person to do something terrible? That's a long stretch.
And, you know, is it, in the end, one has the… the law has the concept that humans have free will?
Absent that, I mean, that's the whole, sort of, complicated ethics of, you know, insanity pleas and things like this. It's like, I didn't have free will. I was forced by a demon of some kind to do what I did. It's not my fault, so to speak.
But most of the time, we kind of assume that humans have free will to do things. So the fact that an AI said, you should do this, it encouraged the human, it gamed the human to get them to do it, but then the human went and did it.
most… the traditional view of, sort of, ethics and the law has tended to be, look, the human had to decide to do it in the end. How brainwashed was the human by the AI? And under what circumstances can you, sort of, argue that the human didn't have free will in this case?
So, you know, there are… there are questions like that. And I think…
there are… and so there are yet different questions when AIs are interacting with kids who aren't thought to have, sort of, the same kinds of judgment that the full free-will adult, so to speak, has. What different constraints should there be then? You know, there are a lot of complicated issues around that.
I think also on the input side of AIs, there's the question of, well, you know, an AI can ingest a lot more data than humans can ingest. How much should you give them? And under what circumstances should you give them? What level of private data? To what extent can you expect that
an LLM that learnt lots of things that you wanted to sort of keep secret as complete things, to what extent can the LLM mince them up to the point where you say, well, it's collectively good that the LLM has all these things.
It's okay that, you know, the pieces of mincemeat are not going to be big enough that we'll be able to see something that we shouldn't let people see, so to speak.
And, you know, that's a whole other collection of issues, and I know in modern times, you know, there's the question of, should the LLMs be able to read copyrighted works of various kinds, and

what should they be able to do with them? And this is something where different countries have now come down on very different sides of that issue.
And, you know, there'll be… there already is a fair amount of migration of companies doing things in this location and that location, because the rules are different in this or that place.
And I don't know how much that will harmonize over time. You know, it's… it's often the case that, sort of, if you're, you know, it used to be the case that if you were registering an LLC in the US, you'd typically do it in Delaware. If you were registering, I don't know, I think a tanker's in Liberia and, …
yachts and the Marshall Islands and things like this. These are places which had sort of an incremental advantage, or, you know, cryptocurrency for a while in Zug in Switzerland, or then a bunch of other venues around the world. You know, these are places that had some marginal and not-so-marginal advantage.
for doing these kinds of business in those places, and the same thing might happen for, sort of, LLMs reading copyrighted stuff. I don't know, I think that that's a question for policymakers to decide what they want to do with that. I think it's a… it's a complicated set of issues around, sort of what
You know, to what extent it motivates different kinds of people to have the sort of protection of copyright, to what extent it is for some kind of public good to have the LLMs be able to read this stuff, to what extent it's merely benefiting a few specific companies. It's a complicated collection of issues.
And, you know, I tend to think that there are various possible solutions
That, involve, kind of, machinery that has been, to some extent, constructed in other industries.
although it hasn't always been totally successful there, to do with sort of pooled collections of intellectual property, where particular kinds of things can happen, and where there's some sort of compensation back to the owners in certain cases, but it's complicated, and it's, I think.
the sort of attribution problem for an AI is almost insolvable, because it's kind of like the AI is going to say, and I think the best color is purple.
And…
you know, that is derived from, sort of, the individual contributions of a whole collection of artists and designers and so on, all sort of vacuumed up by the LLM.
And to say, oh, this person was a purple enthusiast, and they were a .01% contributor to the thing that the LLM said, that's going to be a can of worms to deal with.
Let's see… A follow-up. What should CFOs and risk managers consider when evaluating AI tools for adoption?
Look, I think the basic rule with AI is if you've got a function where 80% success is a win, then use AI.
If you've got a function where you need 100% success, don't use AI.
And…
perhaps use AI offline to build something that a human is going to check that becomes your thing that is a solid brick that you can then use as part of your production system and really expect 100% performance, so to speak. But I think you can't, in, …
Anytime you say, I'm going to get that LLM, I'm gonna get that AI to be perfect, I'm going to remove all hallucinations, I'm gonna make it follow facts, I'm going to have it follow this structure that I'm going to define for it, it's not going to work. It's going to have a certain failure rate.

Now, there are plenty of situations in which, you know, a failure rate, like if you're, I don't know, doing sales, let's say, and you're calling customers, some fraction of the customers are going to tell you to get stuffed, whatever happens. If you use AI in that process to pick customers or something like this, it's all good, because
You know, your, your, your…
You're kind of, you're doing something where 80% success is a win.
Or if you're presenting recommendations for something, you know, this is our recommendation for what, for things you should do between, you know, things you should look at or something. Well, sometimes you won't have the greatest product that should have been the one that was recommended, but there usually isn't a right answer anyway. So those are all things that are worth
Worth having in the picture.
When it's a question, of…
it's a tricky thing when you have to start asking… if you start asking about, sort of, corporate operations. So, for example, another thing would be…
Here's a thing that it's worth somebody scrutinizing more carefully.
Here's an accounting transaction that's worth somebody scrutinizing more carefully, says an AI.
Again, that's a… if you're right 80% of the time, that's a win. If you pick the right things to scrutinize, that's a win. If you say, my AI did my accounting for me, it's probably a lose. Probably a really bad lose.
I think that, you know, again, it's this thing, if it's in an advisory role, if it's something where it's… or even if it's something where, sort of, 80% success is good enough, I mean, if you're saying, oh, I don't know, what would be a good example? If you're… if you're doing,
…
well, there are lots of kinds of things that get done in the world where it's kind of just sort of a, let's try and do this the best we can. That's… those are… those are reasonable candidates for AI.
Now, in terms of, you know, what's the downside risk of AI,
Well, you know, there are all kinds of things. You know, you put the AI on the customer service line, and somebody figures out, sort of, how to convince the AI to send them a free car or something. And then, as soon as somebody did that, it's out on the web, and you've got a thousand people, you know, using that same hack to request a new car.
Well, you've got a serious problem there.
I think that that, you know, validating that that's not going to happen is significantly more difficult for AI than it is for more traditional kinds of computational systems. And again, it's a question of, well, what level of risk are you prepared to accept? You know, another question would be.
Let's say that, Well, these ideas of risk like that are more similar to cybersecurity risks.
Where it's not like you can say, you know, if it's a risk of how much is it going to rain, how often is there going to be an earthquake? We kind of know the distributions of those things. They happen at some rate. We don't know when they're going to happen, but they happen at some rate. with some of these cybersecurity things, and I think also with AI, there just isn't enough statistics to know, in the aggregate, what's going to happen.
It's not something where there's a lot of sort of repeated trials. It's something… well, there isn't some cybersecurity kinds of things, but in really big things, it's, you know, there's only one time there's been this kind of

thing that happened. And with AI, there is a risk of there's a this thing that happened, and nobody expected it, and so on. And as years go by, there's more familiarity with what can happen, and it becomes more possible to kind of derive sort of a reasonable estimate of risk.
I think it's an interesting question. I mean, people are offering to sell AI insurance these days. That's a complicated story, and I don't know how that's going to work out.
There's a question referring back to what I was saying about, about AI.
… about, …
regulation of AI from 5. How would you even regulate thinking? Control the training data? Well, no, you can say… I mean, this is something which… it's kind of a not-gonna-work type thing. You could say.
Watch the internal thinking process. Have some kind of mechanistic interpretation of the incoming internal thinking process, and as soon as the thing has the idea of doing something bad, zap it.
That will be the notion. Now, you can do that sort of externally by reinforcement learning, by saying, you know.
And a lot of that has been done, of saying, hey, AI, if you're going in this direction, don't go in that direction, that's bad, go in a different direction. Instead, a lot of that has already been done, and that's… but I think that that, as a thing where you can expect perfection or regulation, I just don't think that's gonna work.
I think the only thing you could imagine regulating, I mean, you could say, okay, you could be pretty funny. You could say, every AI that's going to be deployed in the world has to have read the US Constitution.
Okay?
Every AI that's deployed in the world has to be able to, recite pieces of the US Constitution.
Okay. You know, I don't think that gets you very far.
Every AI that's deployed in the world has to have this special regulator put on it, which, internal to the thinking of the AI, nobody knows what's actually going on inside the AI, so this is a super big challenge, but every time the AI thinks a bad thought.
Internally, lock it down, stop it. You know, that would be kind of the concept. But that's something I think is quite unrealistic to do, because we just don't know what's think… if we could know what the AI was thinking inside, then necessarily it wouldn't be doing rich, sophisticated things of the kind that we hope our AI will do.
So… That, what am I thinking about that?
… Orange asks, how much of your day would you actually want AI to control?
You know, it's kind of like, okay, humans, sit back, relax, your life's on autopilot.
… I, you know…
I like to do stuff. If there are things in my day where I don't have to actually do it myself.
and they achieve the objectives that I want, I'm very happy to have them be self-delegated. That's been sort of the story of my life for a long time, building automation and tools to make it possible
to just take ideas I have and turn them into real things as automatically as possible. So, for me, anything that does that, takes ideas I have and turns them into real things, that's a, you know, that's a win. If it's a,
I think back in, what was it, a Charlie Chaplin movie from a long time ago, there were things like eating machines, which was sort of a machine to eat for you, whatever that means, because eating is a thing we do for ourselves, so to speak.

It doesn't really make sense. You could have a machine that leads your life for you, but that doesn't really make any sense.
It's a machine that amplifies the things you want to do and actuates them in the world. That makes perfect sense. But when it comes to doing the things I would… the things that, sort of, I want to
Do in my life and get out of life.
and have the machine just do them for me, I don't think that really makes sense. I mean, it's sort of a similar thing for students and so on, when it's like, look, do you want to, … you could just get a machine to do your homework for you.
And clearly, the point of doing that homework was supposed to be that you would learn about something, and you'd get practice doing those things, and so on. But, well, no, you could just say, let me get the machine to do it for me.
And then, what do you have at the end of that?
well, if your goal was just get through the class, get my certificate type thing, well, fair enough. But, you know, it won't have done anything. You know, there is… there is presumably a value for us in things that we internally experience, that we internally have, have done, and that's something that,
You're not getting, if the machine is doing it for you.
But I think the thing about, sort of, the machine is the exoskeleton that kind of, lets you,
Let's do, … …
lets you achieve a lot more than you could otherwise achieve. That seems like a win.
…
Well, there's a question here from Juan. If AI starts doing most R&D, how do humans keep innovating rather than just reacting?
The fact is, you know, I've done lots of R&D in my life, and the vast majority of it I have done with pretty powerful tools that I happen to have been building for a long time.
But it's sort of, you know, is it me doing the R&D, or is it my tools? Well, it's kind of me as, you know, behind the curtain, so to speak. There is a me there that's telling it, go in this direction, not that direction. Then it is a big industrial machine that grinds into that direction much more effectively than I could possibly do it.
But my little go that way, rather than that way. Invent something about this rather than that, and try this kind of invention, and so on. That's me, and I think that's a sort of a surviving part.
And I think that, you know, what happens for me, for example, is I do a lot of computer experiments, and I have to have the idea of what the experiment should be. Then I do the experiment. And then, yes, I have to react to the experiment. That's what, for example, experimental scientists are doing all the time, if they're doing an honest job, is
the data does what it does, nature does what it does, they're reacting to that. And it's the same thing when you're doing computer experiments and so on. You're reacting to what you see there, but yet the kind of the spark of, well, what direction are we going to go in? That's a human thing. That's where the kind of human innovation comes in.
Let's see… …
I'm going to maybe take only one more question here. Boy, this video is very strange. I wonder if somebody…
It's very, very strange why this thing keeps on locking up, but then unlocking.
I don't know.
… Okay, maybe one more.

For today, from almost… says.
I'm going to graduate this year with my CS degree. When I started my program over thousands of highly paid programmer jobs. I'm seeing the job market shrink drastically. What can I do in the next year to better my chances of getting a job?
I think I've talked about this a bunch of other times.
The fact is, you know, The idea of, we're just gonna write
We're just gonna grind out software with software engineers.
It's kind of been a bad idea for a long time.
I mean, people like me have spent huge effort to automate, going from the idea you have about what software should do, to the actual doing of the thing.
People, for whatever reason, don't want to do that. Well, plenty do, plenty use our tech, but plenty don't, and instead choose to spend, you know, a month
writing some big, long blob of programming language code to do something which in Wolfram language would have taken you maybe an hour to program. It's like, somehow, it's… it's… I don't know what it is. I think what it really is, is people were trained to do this really grindy software engineering work.
That…
actually is perfectly automatable, and we've automated it. But that wasn't the job they were trained to do, so the job they do is the job they were trained to do, which is this in the trenches grinding.
So, the in-the-trenches grinding is getting less and less relevant, as there's more and more automation, from whether it's things we do or from LLMs, sometimes that automation is more aspirational than real, but, …
in, you know, so grinding software engineering is probably not the right thing. The thing that is the right thing is taking
computational ideas and applying them to things, figuring out how you take, sort of, a knowledge of how to think computationally and expect to apply it in different areas. So, the jobs to apply for aren't necessarily… aren't the, you're going to be a programmer in a cubicle somewhere.
They're more the, we're trying to figure out how to, you know, automate this process. We're trying to figure out how to make our company run better, or whatever.
These jobs may not be advertised yet, but it's kind of a thing where there's… to know how you can kind of take computation
And apply it to, … to make To… to improve things.
that's an important skill, and that's kind of like inventing the architecture of how you should use computation to do something, rather than deep in the trenches, you're writing Java classes or something. It's, you know, that… that type of thing is definitely not…
You know, that's not on the growth, on growing. It's, you know, it got overbuilt, it got over-educated for, it's not the surviving thing. The surviving thing will be, how do I take
this company, process, whatever, and think computationally about it, and figure out a fantastic new way to do these kinds of things, new set of things that can be done using computation.
That's the kind of thing to learn about.
I have to say that our technology and tools have gotten used for decades to do those kinds of things, and perhaps are the… probably are the leading way that those kinds of things, sort of intellectual leadership kinds of things happen, and learning our tools and the kind of computational thinking around them

is a winning thing to do, even though you won't see as many job ads saying, you know, looking for Wolfram language programmers versus, you know, looking for Java programmers or Python programmers or something like that. Those
those jobs are dwindling, and will dwindle further, but the use, you know, invent a way to make this… be a computational architect, invent a way to make this thing computational, that's just going to grow and grow. And, you know, being able to present yourself
as being able to do that, good thing to do is do projects and so on, which can demonstrate what you can do, and then when you start applying for jobs with companies, sort of figure out enough about those companies to be able to say, here's a thing I could do for you, sort of using computational thinking and so on.
So, … the, …
It's, so, I mean, that would be my point of view. It's kind of think computationally rather than grind out code, so to speak. That's the direction to go in, and that's what will be extremely valued by companies and so on.
And, you know, it's… at any given time, there's always the… the magic, you know, the most valuable person in the company is… switch it around, CIO, CFO, you know, CTO, CEO sometimes. They sometimes get a… don't… they've had their moments in the sun, but, you know, people… the most valuable person, or the chief AI officer, or whatever it is.
There'll come a time where, sort of, the chief
I don't know what it'll be called, computationalist, the chief computation officer, ends up being The most valuable member of the executive team, and, you know, be there, so to speak.
Alright, I should wrap up here. Thank you for joining me, sorry for the various glitches, and … by… well, let's see, I think I might try and do another livestream on… Friday.
But after that, I went back in my natural habitat, and hopefully glitch-free.
But, thanks for joining me today, and, talk to you another time.
Bye for now.