Hello, everyone. Welcome to another Q&A about future of science and technology. These are always fun for me, because

the questions you guys ask get me to think more about the future, which is interesting.

Let's see, we have a question. Here we have a bunch of questions saved up. We have one from anon

asking, can you see optics being used eventually to do what quantum computing claims it can do. Oh, boy, well.

1st thing, let's talk a little bit about quantum computing, I mean, in a strange turn of history. I I worked on quantum computing in the early 1980 s.

Actually in part with Dick Feynman, physicist who was interested in such things. And I realized that I realized recently that I wrote a kind of referee report on the sort of the 1st paper about quantum turing machines, telling the author that I thought that paper was extremely hard to understand, and suggesting a better way to describe what was going on

which the author ignored, and that was sort of lost forever. It's a pity that my explanation of quantum curing machines didn't make it out into the world in 1982 or so, but I didn't think it was that interesting?

Well, quantum computing, then kind of the idea of let's see whether we can do computing kinds of things at a quantum level

became, you know, investigated. David Deutsch, for example, started studying quantum gates and quantum circuits, and so on. And then

in. That was sort of an interesting thing. It kind of had relationships to quantum logic, sort of generalizations of logic that related to the formalism of quantum mechanics. Things like this.

Then the real boost came with Shaw's algorithm in the early nineties. I guess that made the claim that a quantum computer could factor numbers faster than a classical computer could.

I think the big picture that has been somewhat obscured by the formalism of quantum mechanics. But the big picture about what a quantum computer can do is

it is sort of core to the idea of quantum mechanics is that many things are happening in parallel, but we only get to observe sort of the aggregate probabilities of what what took place.

and so that many things happen in parallel gives one the possibility of doing computations more efficiently like. If you're factoring a number in parallel, you could try dividing the number by lots and lots of different possible divisors. And normally in the in the world, as we normally think about it in classical physics. You'd have to do all those trial divisions sequentially. But in quantum mechanics you can get to do all of them in parallel.

The thing that's the big

sort of problem of quantum mechanics in terms of its use for computing is that yes, all those things can happen in parallel. But we want to know the answer, and that requires somehow knitting together all of those different threads of underlying quantum history into something that we can say. That's the answer.

In the formalism of quantum mechanics from the 19 thirties, and so on. This idea of measurement was just a mathematical operation. Just yes, you have all those threads. You have all those superpositions of possible states of the universe, and so on. And it's then just the measurement operator is applied. A mathematical operator where it projects it into a definite answer.

Question is, how does that actually work in practice? In practice? That's the thing where typically you're taking a little tiny quantum effect, and you're somehow amplifying it to the point where we humans can sort of sense it where we can actually do things with that with whatever it was. That was the sort of outcome that came from that little sort of quantum effect underneath. Well, this process of what does it take to do? Measurement has never been properly formalized. We've been trying to do it a bit in the context of our physics project lots of progress actually happening in various directions there. But the main point is that yes, you can work out what will happen in principle, with sort of measurement as a mathematical operation. What will happen in practice is a much harder problem.

And this is one of the things where you know, how does that show up? When you actually make a quantum computer. It shows up usually in the fact that you can't just can't quite get that answer. There's too much noise. There's the phenomenon of decoherence where you sort of think you're maintaining a quantum state, but it's decaying with time.

These are things where people keep on pushing. Oh, we're going to use error correction. We're going to gradually improve kind of the the extent to which we can get an answer from this sort of underlying kind of many threads of possibilities that exist sort of in the, in the, in sort of the underlying physics of the quantum mechanics.

This problem, the problem of sort of what does it take to get an answer? It's really not a solved problem. It's to some extent it seems like an engineering problem. It seems like the kind of thing. Where, if you just were to, you, think about a telescope or something, you know, polish the mirror better than the telescope would work better.

But I don't think it actually is that way. I think it's not really just an engineering problem. My own guess is that there's sort of a fundamental physics limitation associated with the the kind of this measurement process in quantum mechanics possibly related to a thing we call the maximum entanglement speed in our physics project. But something that's sort of a fundamental limitation on what you can do with sort of quantum measurements of complicated systems.

I have to say, it's kind of ironic that I thought that back in the early 19 eighties, although the kind of ways to think about that and the formalism we had to think about those things wasn't as sophisticated then as it is today. But it's still the same problem. And it's the problem that you find out when you kind of go to the back office when you visit a quantum computing company type thing the engineers are like. Well, yes, we're having some trouble.

Actually, you know, getting the performance we want to get. I think that's a fundamental physics problem, not a sort of engineering, you know, lack of engineering effort, you know. I don't think it's the case that, like in the semiconductor industry, where there's just been sort of progressive improvement in how small you can make the features on chips, how fast you can run chips. Well, that's
somewhat peaked out, but the
there's sort of a bunch of engineering improvement, with some occasional physics added, but mostly it's engineering, I think, in the quantum computing case there's a bit more that's really a physics limitation more like when you're building telescopes. The resolution of a telescope is ultimately limited by the physics of diffraction, not by how well you polish the mirror and the telescope.

So where does that leave sort of what will happen with quantum computing. I think that the the kind of the effort to study the physics of quantum level processes is very worthwhile. I think that the formalism of quantum mechanics, of thinking about doing computations in parallel, and so on, and the quantum version of that that's also perfectly worthwhile and useful. I think that sort of having the big banner of we achieved. Quantum computing is a little bit, you know, is a good marketing message, but I don't think it's reality.

I think that the it's sort of a funny situation right now, because there's ultimately a lot of energy being put into quantum computing because people are sort of worried. If it turns out that Shaw's algorithm can suddenly factor numbers really efficiently break out of the almost exponential amount of effort that it seems to take to factor numbers. If you can break out of that, then all the number factoring based public key cryptography in the world will crumble in a day when somebody makes that quantum computer that can can factor numbers quickly in a routine fashion.

The question is that going to happen, or is it like saying, Will we invent, warp, drive, and be able to, or, you know, go faster than light, and zip around, and and not have to invest in airplanes, because we can just zip from place to place faster than light, or whatever.

I think it's 1 of these things where my own guess is that it really isn't a thing that's possible, based on kind of the way that we are in terms of what we want and definite answers from from these systems that are sort of following many threads of history, and so on. But I think what sort of survives from the whole

kind of effort to investigate kind of quantum computing. One thing that survives is the possibility of radically new physics, technologies for implementing computing. I mean, we know, and even my own efforts and the principle of computational equivalence, and so on.

Give one a great deal of confidence that computation is possible in many kinds of systems. It's not just semiconductors and electronics that do computation. It's molecular scale systems do computation systems where you can imagine doing computation with fluids. You can imagine doing all kinds of computation. It's sometimes a difficult problem

to kind of set up the initial conditions for the computation and read out the results. But many things in the natural world in the physical world compute, and the question of which ones are convenient to use technologically is one that is sort of an engineering problem.

and a lot of effort has been put into quantum computing of investigating different sort of physical substrates for computation. And that seems like a worthwhile thing. And it's whether those computations are operating right down the level where you have to be thinking about quantum mechanics, or whether they're just operating at a scale that's smaller, faster, whatever than semiconductors. The latter is enough to be very interesting.

So one of the places where that comes up is optical computing whether one can use photons rather than electrons as a sort of substrate for computation. I mean, it's worth realizing that there are a variety of substrates for computation.

That sort of work in a little bit the same way, like in biology, we can think of sort of flows, of molecules, flows of ions. Things like that as well as flows of electrons as being things where biological systems are sort of computing at a molecular scale. But the substrate that they're using is more like molecules than it is like specifically electrons.

but the other possibility is using photons as the substrate as the sort of carriers of information in computation. Photons are used all the time as carriers of information in lots of situations, most notably in things like fiber optics.

That is sort of the standard these days for communications, whether it's the Internet, or whether it's in server farms or whatever else that's kind of the way information is transmitted is with photons going through glass fibers for fiber optics.

But the question is, can you compute with this? Can you make those photons not just be transmitted from place A to place B, but actually interact in some way that does a computation that's difficult.

One of the things that made electronic computation possible was the discovery that you could make electronic switches. First.st Those were vacuum tubes, later transistors. But that's what you know. Microprocessor is billions of little switches made up of transistors of field effect transistors implemented with semiconductors. That's sort of a big question. Can you do that with photons? Can you make an optical switch?
Where? And sometimes, when people use the term optical switch. They mean something where there's electrical input and it switches whether whether a photon can go through or not. What I'm talking about here is a photon switched
sort of gate that determines whether a photon can go through.
We don't know how to make those yet.
We know for very intense light. There are ways to make switch like things, but it requires that you're sort of pumping so much light into something that you're saturating the amount of light that atoms can absorb, and so on. And that's far. If you had to do that, you couldn't make something small and efficient, and so on. So it's sort of a. The unsolved problem of optical computing at some level is is to make kind of a photon to photon switch.
If that was done, then immediately you could start switching lots of technology from from electrons to photons.
Now, you know, is there fundamentally quantum mechanics there, we're used to photons having interference patterns and so on. We're used to kind of the wave character of photons. We're more used to that than we're used to. The wave character of electrons. Electrons as a result of quantum mechanics also have a wave character
at the level of the actual way that microprocessors are made with feature sizes of a few nanometers, and things like this, the wave nature of electrons does start to matter, and certainly in the etching of semiconductors it matters a lot that there's that there. Well, at least with electron beams, and so on. It matters that there's a wave character of the electrons and such like.
but in in terms of sort of the
the sort of the fanciest effects of quantum mechanics. Well, as a practical matter, when you figure out how to make a microprocessor, you do have to pay attention to quantum mechanics in several different ways, in the ways that the actual semiconductor works in the ways that you're etching things out and so on and in the way. But you're not having to pay attention to it at the level of kind of thinking about. Oh, I'm doing a quantum algorithm rather than a traditional classical algorithm with classical logic, and so on.
The same will be true of photonic computation. There's a lot that will be once one can do that. If it's even possible to make photonic switches. The you, you can.
you you would expect to sort of implement the same kinds of things you do in in classical computers and classical logic.
Could you make use of the wave nature of photons and sort of interference patterns, and so on. I'm not sure. Probably. Yes, probably using some of the same formalism as in quantum mechanics, although the effects don't have to be sort of the the really deep effects of quantum mechanics.
I mean, it's
It's worth saying, by the way, that there's sort of a different issue, which is when you're dealing with a microprocessor, and you have electrons going from here to there they go at a certain speed they go at. Oh, I don't know how fast it is maybe a 3rd of the speed of light or something. If you have photons well, photons in a piece of glass, go at 2 thirds of the speed of light. So

you know, there are places where it it matters to have things, signals transmitted optically rather than electronically, although that's not as dramatic as an effect as to have sort of the whole thing operate photonically.

But I think one of the things that I've suspected I suspected for a while was that sort of eventually quantum computing would sort of turn into optical computing, and then sort of victory will be declared. Type thing. I'm not quite sure that that's quite what's going to happen at this point, because I think some sort of basic ways in which optics can be used in computing those already happening. The sort of the problem of inventing the photonic switch

is still out there in the future and might not even be possible. I mean, it requires that there be some strange kind of material that will deal with individual photons, and will sort of have this switching behavior. One just doesn't know such a thing that such a thing exists, or even could exist.

You know I have to say in terms of of quantum computing, and so on. I think, sort of in the quantum realm. The thing that seems to be the most immediately useful is quantum sensors where one's using effects that are from sort of

foundational quantum mechanics to make very sensitive measurements of things. Most obvious thing is magnetic fields where there are sort of flux quanta that you can have, where where those are sort of the very lowest level, the very smallest units of kind of magnetic field that are sort of quantized. And you can start measuring those those quantum flux units effectively, and that allows you to make a very sensitive magnetometer.

There may be many uses for that. Certainly some that have already come into existence are things like the earth's magnetic field changes depending on where you are. And so you can use that as an alternative with GPS as a sort of local way of measuring where you are on the earth's surface.

And there are things one can imagine with MRI, for example, other kinds of places where where actually not so much that maybe, Meg, magnetic encephalography, where you're measuring magnetic effects of nerve cell firings rather than electrical effects, and so on.

There are all kinds of places where you can imagine that very sensitive magnetometers might be important. I don't think we know yet what all those places are, but that's an example of something that seems to be coming from sort of the quantum realm.

And there may be other kinds of very sensitive measurement techniques that can be used, that sort of come out of quantum effects and so on.

So a few thoughts about that.

Let's see.

Well, there's a question here.

Okay, let's let's look at a few of these about quantum computing. Hannah, do you think quantum computing will actually change everyday life? Or will it just be for specialists. Well, right now, it's not changing anything, I mean.

in sort of the spin-offs, from quantum computing of of sort of better devices for doing computing will be

probably incremental change.

probably just sort of slowly introducing that technology in place of existing microprocessors. I mean, it's conceivable that there will be a big jump

that suddenly it will be possible to do computing a thousand times faster or more. I'm I'm don't immediately see that I think it's more likely that we'll be able to do computing at a smaller, more molecular scale, more like the way biology seems to do it. And well, biological. Some aspects of

biology, I mean in brains. That's not what's happening. There are great big neurons in brains, but at the level of molecules, the sort of dance of molecules that makes up life.
That's
more like actual computing done at the molecular scale. But in a sense, what's happening there is, it's computing. But we don't know what it's computing other than sort of
keep, you, you know, alive and and happy, so to speak, because it's sort of a dance of molecules that are achieving things that we can simulate on a computer and say, we're doing computing there. But what if we say, I want that dance of molecules to compute this math problem that I'm trying to solve. We don't know how to plug that together yet.
But
I think it's an interesting question. If computing suddenly got a thousand times faster, what kinds of things could we do? It's been. It's been interesting over the course of years. I've
been doing language design. Now for what? 45 years, or something, and the things that you can sort of take for granted that computers will be able to do have moved a bit over those years. I mean, the speed of computers has changed dramatically over those years. Let me give you an example. If you have a user interface.
And you say, can I? How much can I compute? Can I just write this complicated program and have that whole program run in the time it takes me to notice that I move my finger on some slider. Well, the answer is, one can do that now. That was too heavy a lift.
let's say 40 years ago. And so that's something where sort of faster computing makes that possible
video conferencing. For example, another case where sort of there's a there's a in a sense, there's a a fixed time horizon, a fixed time interval that's determined by our ability to notice whether a video is whether something is really moving in a video, maybe 1 15th of a second something like this. And the question, then, is, is computing fast enough that in that 1 15th of a second we can transmit that video frame, and so on.
And the answer is, there came a time when that happened. There came a time when we sort of were able to get realistic video transmitted, encoded by a computer transmitted and so on.
I mean, it's the same kind of thing that happened well, in audio, much earlier in video. Now, I suppose the next kinds of things are synthesizing video in real time, being able to use kind of AI techniques and others to kind of, do you know, create the scene in real time? Already this is happening a whole bunch in video games. But there's more of that that one can imagine happening. I mean, there's a lot of things where it's interesting to wonder what what one would be able to do if
one could compute a thousand times faster.
And you know what kinds of things were things that well, even today there are so many things that get done by computers where in the past it would have been. You couldn't use a computer for that. It would just take it's crazy to to waste that much computation. I mean the typical computer these days. Most of what it does is be idle, and it's running an idle loop.
And you know, in the past, when computers were much younger. It's like every moment of computer time has to be used for something. It's not
the concept, or even even the concept that you can watch a video, a movie on your computer would have seemed crazy. But until it became the case that it was so cheap to do computing that you might as well just have that video playing on your computer screen, not on a custom television or whatever. So it's an interesting question what else we could do if I think

sort of a lot of figuring out of predicting what's going to happen in the world that might become possible. And when I say, predicting what might happen in the world, I mean at a very local scale that might become possible if you had a lot more compute at your disposal, would you be able to do? I mean the kinds of things that are now done with things like machine learning training and so on.

Yeah, those would become faster to do. And it might mean that you could do more of traditional current machine learning training on the fly. I don't happen to believe that the current huge investments made in machine learning will still need to be made when we figured out more about how machine learning really works. But even if you assume that, that's what you're doing, being able to compute a thousand times faster brings into kind of

personal range some of the kinds of things which in the current time are only accessible to sort of large companies spending with huge investments in these kinds of things.

Let's see

Pedro asks, what could quantum computing bring to AI and machine learning?

Well, 1st of all, as I keep on saying, I don't think the big flag of quantum computing is going to work out.

But the ideas of quantum computing? That's an interesting question. And you could say theoretically, if quantum computing happened, you know what would, what would occur. But, by the way, coming back to the previous question about quantum computing, changing everyday life, you know, if it really slam, dunk, worked in a way that I think it won't

then sort of the way cryptography is done would change. People are talking a lot about quantum safe algorithms. I think this is a very, you know, difficult thing to discuss, because it happens that there's this one example from the 19 nineties of a cryptography related algorithm, which is sort of crushed theoretically by quantum mechanics.

And you could say, Well, what about all the other things? They're not that, many of them, but all the other kinds of methods that people have invented for doing public key cryptography, for example. Well, we don't happen to know that those can be crushed by quantum computing, even if we had the theoretical, even if the theoretical side of quantum computing worked out the way that the theory

kind of indicates that. Well, it doesn't really indicate that it does it. The theory, if you just believe the theory, which isn't the thing that relates to reality. It's all rather tenuous, and I think that the question of whether there are sort of quantum safe

cryptography. Again, it's a driver to study different types of cryptography using. I don't know lattice methods or something rather than using factoring integers. You know, using questions about whether different kinds of mathematical problems.

and you know that's a perfectly reasonable thing to study the fact that you say this is quantum safe. I think that's pretty tough to say.

But you know, what effect would it have on the world if somebody made that practical quantum computer? And it, you know, the day it happens, you know, lots and lots of sort of computer security would just fall completely immediately. It would be replaced by other methods. That sort of don't seem to be as attackable by quantum mechanics. If such things exist.

Let me let me just make a few technical points here. So the the big question is related to the P versus Np problem of computational complexity theory. The basic question is, if

you can, there are computations that you can do fairly quickly, like, if you've got 2 n digit numbers, you can multiply them together in a time. That's about N, it's a, we know. N, log, log

log n multiplication, speed things for large. N, so you can. You can multiply numbers very, very quickly
if you
If the numbers have n digits, you can multiply them in a time. That's about. N,
if you ask, can you factor the number? The most obvious algorithms involve sort of testing every possible number less than, let's say, the square root of the number, and the size of the number is about 2 to the N exponentially larger than the number of digits in the number. And so that means that that kind of strategy means that you will take exponentially long to do that factoring. Now, can you prove there's no better algorithm for factoring? No, you can't. Nobody knows how to do that.
But, for example, testing, if a number is prime, there are actually fast algorithms that are of order. The number of digits in the number, not the exponential, of the number of digits in the number to test whether a number is prime, but to factor a number. So far as we know, it's still roughly exponential in the number of digits, but proving that there's a lower bound proving that there's no way to factor any faster than that. That's something nobody's been able to do.
There is this notion of problems. Well, like factoring is an example of a problem where once you've guessed the factors of the number, you can easily multiply them together and check that they're right.
But the question is, how do you find those factors in the 1st place. And so one talks about 2 classes of problems, P problems, so-called polynomial time problems where the difficulty of solving the problem goes up only like a polynomial in the size of the problem. So things like multiplication of n digit numbers. It's a polynomial. It's of order, n, the time to solve that problem. On the other hand.
factoring is an Np problem, a problem that's a non-deterministic polynomial time problem. If you could guess the answer, then you could check it in polynomial time. The question is, can you find the answer? The P. Versus Mp. Problem is the question, can you find the answer? If you can check it in polynomial time. Does that mean? You can also find it in polynomial time?
Okay, so that's sort of the definition of that problem. It's sort of a core problem in computational complexity. Theory is the class of problems that can be done non-deterministically in polynomial time is that there are problems that are in Np. That are not in P. That can be done non-deterministically in polynomial time, but not deterministically in polynomial time.
And what was discovered in the early 19 seventies is this notion of Np completeness? There are problems where, if you can do, where there are problems which are sort of equivalent in difficulty where you can translate between them with only sort of polynomial translation. So, for example, things like various kinds of graph coloring problems.
various problems about whether logic expressions can be satisfied, all sorts of kind of routing problems, and so on. These are all examples of so-called Np. Complete problems. If any one of those problems could be done in polynomial time. They could all be done in polynomial time, because all those problems are complete for the class of Np problems you can emulate, you can encode any Np problem in terms of one of those problems.
It's very similar to the thing that you can do with universal computers where you can encode. You can write a program for one computer that will emulate the behavior of another computer. The Np case is a more junior case because you're not doing arbitrary computation. You're just doing these non-deterministically polynomial time computations. So there are class of problems known to be Np complete. A whole bunch of them. Factoring is not one of them. We don't

actually know whether factoring, whether, if you could solve factoring, it would allow you to solve other Mp. Problems

if you could. If if there's a chain of ifs.

if factoring could really, if you could build a quantum computer and it could then do factoring fast, then factoring would have sort of fallen below the. And then the question is, Well, what does that mean for the other Np problems? Because basically all public key cryptography is based on the assumption that Np problems are hard and in general.

And so there's sort of this question of well, what gets left. If factoring falls.

you know, does it? If if an Np complete problem fell, they all fall.

If factoring falls, it probably doesn't pull very many other problems with it, and certainly it better not pull any Np complete problems with it, because then you would have pulled all the Np complete problems. So that's kind of the this question of sort of what what happens? You know the mythical day where quantum computers break

public key cryptography, factoring, based public key cryptography, you know. Does that just mean everybody just migrates to another Np problem, maybe an Np complete problem. That's been a difficult thing in cryptography

to find a case where you can base the security of cryptography, not just on an Np so-called Np. Hard problem, but actually on an Np complete problem that hasn't been successfully done. We don't know how to sort of peg cryptography, the security of cryptography to

complete Np problem. We only know how to do it to Np problems that aren't part of that, that pool of Np complete problems.

Okay, anyway, long answer to that. But I was coming back to Pedro's question about could quantum computing. What could quantum computing bring to AI and machine learning? I think one of the things

that happens in, for example, an Llm. You know, the main mission of an Llm. Is. Guess what the next token should be in a stream of of output. So you know, you're always trying to. The Llm. Has been trained to work out what's what's the most likely next word in the sentence. You know the I don't know. The the cat sat on the.

you know, blank, probably matte type thing. It's it's always that. That's the mission of the Llm. Is to is to figure out what what should the next word be

now usually in a typical application of an Llm. There's a whole. It it has worked out

of the 50,000 words in English. Let's say, what are the probabilities that should be assigned each of those words?

And normally it just picks one particular word to use sometimes that so-called 0 temperature. It's picking the word that it thinks is the most probable word usually for doing things like writing, essay responses and things. Llms are set to temperatures of like 0 point 8 or something where they are picking words with.

we're picking somewhat at random these words, with probabilities that are related closely related to the probabilities the Llm. Thought that word would occur. They're not always just saying, pick the winner the most probable one they're saying. Pick a word at random, weighted with the probabilities that the Llm. Assigned to it. But, anyway, what's happening is you've got a single word that is being picked as the next word.

What one would suspect from sort of quantum. The quantum idea is. Well, don't just pick one word. Look at sort of a whole range of possible words. Do the thing that we're describing. Quantum mechanics as doing. Don't just follow one path, follow a whole bunch of different paths that maybe branch and merge, and do all those kinds of things.

and so one can imagine sort of a quantum version of an Llm. In which it is not just giving you that one string of words, but it's treeing out all these possible sequences of words. And maybe it's doing that with some watching for whether it achieved some particular objective.
It's so this kind of sort of the quantum idea could be a. You've got many different things coming out of the Llm. Now, if you had a quantum brain, you could happily be reading off all those different threads of what was being said. The Llm. Would be saying many, many things. It will be speaking in Llm. Tongues, or something in parallel, so to speak, many different things, and your you know, your quantum brain
would be able to process all those different sort of threads of what was going on.
But that's not how our brains actually work. We expect a single stream of words to come out, and somehow that's what we're looking for
now. So then, we might have something where we've got many sort of streams of words that are being generated internally by the Llm. But then they're kind of we're picking the ones that satisfy some particular criterion. It's a little bit like what's happening in factoring numbers where we can imagine having many threads going on there. And we say, let's look for a thread that satisfies the objectives we want.
You know, I noticed with Gpt. 5. For example, there's some capability to make the output follow a so-called context. Free grammar. So that's that's saying, rather than having it be sort of a free form thing where? Yeah, it's roughly right. But the you know, it might not get. If it's writing a piece of code, it might not get the syntax of the code. Quite right.
There's a way of constraining the output to precisely follow. Kind of the grammatical rules that say, once you open a parenthesis. You've got to close a parenthesis once you've done this, etc. Etc. Etc. And
I think the that's sort of going a little bit in this direction of, you know, there are things that can happen. You're you're kind of. You're pruning the thing based on.
You're pruning these possibilities not just based on probabilities, but you're pruning them, based on some structure that you're looking for, and maybe in some cases you have to follow a thread forwards that eventually doesn't work out. You have to backtrack and follow another thread. One could also sort of pre-generate some of those threads, do them in parallel, and then be just throwing away the ones that don't satisfy the constraint that comes from this this grammar that you've defined.
Let's see.
Buff, let's see, following this
evil is asking, what do you think about neuromorphic systems?
I don't know. I mean, that was one of these buzzwords. It's there have been many of these kinds of make
make computing leverage what we know from nature in what happens in computing, I mean in in engineering.
there have been some success stories, partial success. Stories over, told success stories undertold success, stories about where we got a clue from nature, about how to do a piece of engineering, whether it was sort of
Velcro and the gecko feet, or whether it was, you know, aeroplanes and have wings, although they don't work the same way as bird wings, particularly in airplanes. Nature gives one all sorts of clues about how to do engineering.
I think. and the the question.

the whole idea of neural networks, for instance, did come from from 100 years ago, basically close to 100 years ago, did come from sort of thinking about how brains seem to work and then emulating those in what ended up being electronics. And now in software and so on. That was sort of a A From nature kind of idea. I mean, it's worth saying in terms of sort of getting ideas from nature for technology. The place where that happens just all the time

is in life sciences. It's sort of embarrassing. How many of the things which have been invented, whether it's nanopore technology for for reading, for sequencing DNA, whether it's gene editing technology with Crispr and so on these things. We didn't invent these things. We found these things, you know, Crispr is part of the immune system of bacteria nanopores. Oh, gosh! Where's that from?

That's another biological mechanism. But there's a whole bunch of places in things related to life sciences, things at molecular scales where we didn't invent this stuff. We just found it

same is true, with a vast array of the drugs that have been discovered, the few 1,000 drugs that are in out and about in circulation, and so on, that the vast majority of those came from

when they were not designed by sort of explicit human engineering effort. They were things found oh, that's the bark of this tree, and it does this, and etc. Etc. Etc. They were chemicals that had been produced by biological organisms, for example. And then we just sort of brought them in and used them.

So it's sort of natural to say, Well, what can we learn from biology? After all, you know, it's it's been going a while, and it's had this process of adaptive evolution which has got it a long way, and you know, if we, if we look at raw numbers, I think 10 to the 41 organisms I once estimated have lived in the history of life on Earth.

That's a lot of organisms. It's not absolutely out of range of the kinds of things, the number of operations that computers can do, and so on. I mean, if we've got oh, let's say, 100 billion computers in the world.

That's, you know, 10 to the 11 computers doing. Let's say, 10 to the 10 operations per second. That's 10 to the 21 operations per second in a year. That's 10 to the 28 operations roughly. So we're getting up there. And obviously, biological evolution has had a couple of 1 billion years to have all those organisms exist and so on. But you know, but it's also a very inefficient system for discovering things.

What's my point? I mean, in a sense, we can expect that biology has been ahead of us in some ways, but it has many constraints in the way that it builds. Things, you know, proteins are just. Not that they're wonderful and can do all kinds of things, but one imagines they're not the best way to do lots of engineering kinds of things. I mean, I think in my own efforts. I've done kind of the

the sort of pure abstract software analog of what biological evolution has often done. I've been sort of searching these spaces of possible programs. Been doing that for years, finding all kinds of interesting things things that you can. You can, if you just sort of randomly go out into the computational universe of possible programs, you find things that you wouldn't actually find if you had to go down a path

where you're always having to have a successful program, because in biological evolution you don't get to have. If you, the organism die, you don't get to have children and pass on your genetic information.

So it can't be the case. If there was. If you wanted to get to this place where there's this amazing discovery that was made by in in the way that a biological organism was constructed. But to get

to that place you have to go through a kind of swamp of organisms that wouldn't make it. Biological evolution can't do that.
I've actually studied in the last year or so. I've studied a lot about what biological evolution can and can't do different topic. Maybe we shouldn't get into that now. But
back to the question of sort of neuromorphic computing, I think that the issue is to what extent can we emulate different aspects of how brains work and expect that to be useful? It's a key question. What matters in how brains work?
For example, there was a time when people were trying to make proto aeroplanes, and they decided to put feathers on their wings, because, after all, you know, birds have feathers on their wings, and they successfully fly. Does it matter that you have feathers on the wings? Well, it turns out it doesn't matter. You know there are not a lot of aeroplanes that I've ever seen that have. In fact, I don't think I've ever seen an airplane with feathers on its wings. That just turned out not to be a key aspect of the problem of flight.
So in the case of of brains and and kind of computational thinking, like activity. To what extent is, what aspect of human brains important?
Well, what's become clear from sort of the success of things like Llms is a lot. Isn't that important we didn't know until Llms. Came on the scene. We didn't really know it could have been the case that there's some weird physical effects, and so on in brains that we'd never really emulated in electronics. And that were the killer thing we absolutely needed to be able to make a brain-like thing.
I think we now know that that isn't the case. It's sort of all abstract software kinds of issues. It's not some magic new piece of sort of physical hardware.
And and I think that's sort of the the sort of least what I remember of neuromorphic computing. A lot of it had to do with sort of things that were almost hardware tricks that one was emulating from
from biology, whether it was using continuous values for variables rather than the discrete things you have in bits and computers, and so on. And I think the evidence now is that for the things we do that involve sort of thinking like processes. You just don't need that stuff.
So it doesn't seem like it. It seems like it's 1 of these things where you're hewing too close to the actual biology. And you don't need to. And because what you can build with electronics is a bit different from what we can build with with biology, it's it's kind of like, go with the stuff that can be just done with electronics. Just like if you were, you know, building a metal bird
is hard. That's just not what biology does. It doesn't make, you know pieces of aluminum with rivets in them, but it does fairly easily make bones and muscles and feathers and things. And so it's a different, you know, it's a different technology stack. And one should sort of pursue it in a different way.
Let's see,
boy, a lot of interesting questions here.
Prab is commenting. I keep noticing this thing with Llms that they're clumsy with math, even though they keep saying, it's so great I don't know why they say it's so great. I mean, you can easily test it. And it's not, and it's not surprising that it's not, it's it's not for sort of the same reasons that we humans are not. We humans
can do conceptual math. If we really know what we're doing quite well.
We humans are not to be expected to do detailed mathematical calculations in our minds. Just not what we're built for. It's what computers are built for. But it's not what humans are built for, and it's not what neural nets are built for, either.

Let's see.
Oh, boy.
so many
Well, Hilzer is commenting about about things like Chat gpt about running some sort of logic on the input.
Why don't they run the same engine on the output. Well, I think that the input is mostly constraining the kinds of things you can talk to it about. It's sort of a a pre censorship kind of kind of thing. I think the output there's similar kinds of things run. But this question about, you know, is what you're saying does what you're saying make sense. Well, what's the gold standard for that? It's in
for things that involve computation.
The gold standard is
what the result of the computation was. You shouldn't be using a neural net to compute to do that computation. Anyway.
if the question is something which is a more vague kind of linguistic kind of thing it's like, how do you tell that the answer was right. You can feed it to another. Llm, that's what one often does. That's what we do for lots of things we're doing is you ask another. Llm. Did that 1st Llm. Do the right thing, and so on. But I think ultimately the thing that people have been doing a lot.
We've tried some things with this. I haven't worked very well where you're kind of getting
the Llm. To create something which you can then sort of check. Did it meet this framework?
I think there will be better success of that in the future, but I don't think that's the best use of Llms. I think by the time you're trying to make something which has a sort of precise framework, you should be using computation to do that. And I think I mean, this is a longer subject, but it's kind of the whole story of the best thing we can do. Given what we know is to link Llms
to basically the computational language that we've built to computation and to precise knowledge that
that connection of the Llms as kind of linguistic interface layer
connected to precise computation and precise knowledge. That's a winning story.
I think the idea that you're going to have a bigger neural net that will eventually sort of somehow magically reach that level of precision. It just isn't going to happen. And we can see that from things I've done, for example, on trying to understand the foundations of machine learning. But the sort of dynamics of the industry are such that there's a moment where people just don't want to hear that, because it's it's a kind of a complicated flywheel of sort of investment and hope and enough performance to sort of keep things going, and so on. But I think the thing we're observing is that these kinds of things about, you know, can you make a neural net compute? The answer is, no, and it isn't getting any better, and it's not going to get any better. And it's not the right thing, anyway, because it's like saying, let's teach all the humans to do math or run code in their brains. Good luck with that! We humans don't do that.
But you know, people like me have spent a lot of time building tools to let us humans get the leverage of doing that. I mean, it's just like saying, Why don't we just teach humans to fly. You know. Well, actually, that's just not something we can do.
Better use airplanes or drones, or whatever you're using to fly. Don't. Don't try and teach the humans to fly. And I think it's the same kind of thing with neural nets. You can. You know, we don't teach humans to run code in their brains. There's no class that teaches people running code in your brains. Similarly, we don't get to teach neural nets how to run code in their brains, because it's just not the kind of

that a system like that does. However, we can make great use of the more precise, more engineered kinds of things that like I've spent the last. I don't know 45 years building
to supplement that and achieve things that would not be achievable with the pure sort of brain-like neural nets on its own.

And that's surely the future, although it's taking a bit longer to to arrive in in in a kind of strong way than than I might have hoped.

but the future always takes a bit longer to arrive than one hopes. I think it's been very obvious since. Sort of pretty much the day Chatgpt came out that this is sort of dynamics one should have, and we had built versions of this within within a couple of months of Chatgpt coming out. But it's still not
a sort of a clearly mainstream thing. For for various, I think industry, pathology reasons right now, and I suspect that will eventually change.

Let's see.

Oh, boy, so many interesting questions. Okay, there's a question different direction from test will diy genetic engineering be popular in the next few years, eg. To grow stripy carrots. You know I have thought for decades that the time when I really know that we have understood. Sort of genetic engineering is when the fruit you buy in a grocery store already has a barcode on it that grew with the fruit.

Okay, that was a silly thing to think about, but I had sort of wondered that like 30 years ago, or something. And you know, instead of anybody having to put that sticker on the on the peach, or something. Just have the peach naturally grow a barcode that says kind of I'm a peach. Now, of course, in modern times image recognition could perfectly well establish what kind of fruit it was. I didn't think of that 30 years ago.

The

You know how far will sort of cute, you know. Genetic engineering for these kinds of things go, you know, I recently bought tried to buy a glow-in-the-dark plant that has jellyfish
genes in it.

My assistant, who took delivery of this thing. It died in her care. I never saw it, she said. It was very dim. It didn't really. You had to be in an incredibly darkened room to see this plant glow in the dark, but I've always thought that things like sort of the test of weird.
genetic modification. Fashion is when humans start modifying their genomes to have sort of glow in the dark features, and so on for themselves. I think there are lots of weird things like that one can imagine.

It is not a trivial problem, because going from the genetic program of an organism to the kind of well glow in the dark is easy. That's just add Gfp protein to things, but doing something like putting a barcode on the peach or stripey carrots, or whatever. These are not so easy, because the challenge is to go from the program that is on the DNA. To what's that going to make?

It's something where people had imagined I don't know 15 years ago or something. Now, when when the Human Genome Project got done that we'd be able to sort of get our programs and then immediately read off all these things of medical significance or other kinds of significance directly from our genome. The main message has been, things are more complicated than you can possibly imagine. You know. I was actually happened to just be looking recently at just genetic
encoding of eye color, which I thought was a fairly straightforward trait that would be determined just by one kind of base pair or something in the genome. No, it's actually much more complicated than that. And that's been the story of almost everything. No, it's much more

complicated than that. So that means, if you say I want to make this thing end up with stripes, there will typically be this very complicated dance of different genes interacting with each other that will lead to the thing having stripes.

There are cases where one can make one simple modification and something happens. Something goes wrong, but I think the more general case, if you're trying to arrange to have it have some particular form, it is going to be a complicated dance of many genes, and we don't really completely know how to do that. In fact, some of the work that I've done, and sort of foundations of biological evolution and foundations of medicine, looking at kind of simple computational models of these things makes it clear. Why, that's hard.

If you, there are certain things where you can, you can sort of artificially evolve. You can change the program by sort of artificial evolution to get to the point where it does some kinds of things. But if you say I want this pattern to have this very complicated form that has these specific characteristics, it's hard to even get to that, even with artificial biological evolution.

I mean, it's a the reason biological evolution works at all is that the fitness, criteria for what will survive are fairly fairly course, they're not things that say, in order to survive as a plant, you have to be able to compute the 1 millionth prime. They're much sort of

coarser kinds of criteria. And I think if you, if saying I want the thing to have a particular shape, or something, or a particular pigmentation pattern, that's a pretty difficult thing to achieve. And you know the the idea that you can just go in as a sort of genetic designer and figure out how to get the organism to look like that is not is not yet

reality, and I think it's going to be a difficult thing. I think that there will be particular sort of engineering pathways that will be understood for doing particular things, and then a lot will be kind of computationally irreducible in between. I'm kind of reminded of the 1st blade Runner movie

that I think opens with a haven't seen that movie since it gosh! Since probably since it came out. So my memory of it is a bit dim, but I seem to remember the beginning of it. Features a genetic designer who has all kinds of little devices in his house for a creature that opens the door, a creature that does this, a creature that does that?

That's an interesting vision.

I think we're still quite a long way away from making that reality. And I think that will happen through having particular mechanisms where we can understand how to do the engineering. And then a lot of kind of computationally irreducible stuff sort of filled in around the edges, so that that thing that was supposed to open the door might usually do that, but it has enough sort of computational irreducibility in around the edges that it might also, you know, jump on the table or something instead.

I think

the yeah, that it's an interesting thing back. I don't know. 40 years ago, or something, maybe close to close to 50. There were people who were confidently saying, You know, this computer revolution, that's that's nothing. Just wait for the sort of biological revolution of things that one will be able to do by sort of creating custom, biological organisms, and so on. That's

that hasn't really happened yet. I think that the place where that becomes important is that sort of making.

since if we're interested in interfacing to us, we are biological organisms. And so the interface to us has to be, in a sense biologically aligned. And and there, I think we have more potential for sort of biotechnology of that sense in that sense to be important.

Let's see,

leftist commenting about growing neurons for a computer. As a company, we've interacted with a bit that's growing brain organoids and trying to have them service as computer components. It's really a challenge to communicate with that brain organoid and to train the brain organoid to do what you want. And I don't know. I think it's an interesting thing, particularly for neuroscience research to be able to grow and do things with brain organoids. I doubt it's going to be a competitive thing for actual computing anytime soon.
Let's see.
Hill's comments. I've seen a lot of Phd. Students use Chatgpt to for review my paper, and the comments are similar to what a real human would give.
Yeah, I mean, insofar, as commentary is sort of
based on sort of the average of the population. I think that's what you're likely to get from an Llm. If you want the Llm. To suddenly come up with a brilliant idea. Oh, you missed this thing in your paper. You should have done that instead. I don't think the Llm. Is going to be the winner for that.
Let's see.
Nancy asks, is there a limit to automatic design of algorithms?
Well, that's a complicated question. Because
what does it mean to design an algorithm?
So you have to know what the argument is supposed to do.
So you have to have some specification language. You have to have some description of the algorithm
that already exists. Now, the question is, you're asking a little bit if we can say, I want an algorithm that achieves these objectives. It's a little bit like the Np problems I was talking about earlier. We want to find a way to achieve this objective. But I'm not telling you how we're going to do it.
Then you know to what extent can we fill in the algorithm that you need to be able to do it? That's sort of an art that hasn't really developed that far. So, for example, you might say, let me give you a thousand examples of what I want to have happen. Now make an algorithm that will generalize that neural networks
do that, but they do that in a sort of fuzzy way, for kinds of things that are a little bit like what we humans do things things like that, for they don't. If you say here are some examples of factoring. Now you know, generalize that and be able to factor any number. It's not going to work.
How do you get kind of can you fill in? Can you automatically discover an algorithm for factoring?
But something as sophisticated as that. We absolutely do not know how to do that
for something like sorting. If you give me a fixed number of objects, let's say 16 objects.
And you say you've got some something where you're just going to compare. Object one with object 7. Object 8. With object, 11. Object whatever with whatever you're doing a sequence of pairwise comparisons.
That's a case where it has been fairly successful to find optimal sorting networks just by searching the space of possible sorting networks. I think. What a couple of things that are interesting about that the sort of engineered algorithms for sorting, if you looked at sort of the pattern of the comparisons they make. It's a fairly regular pattern.
But if you just go search the space of all possible sorting algorithms, you can come in for a particular number of things you're trying to sort and say, this is the optimal sequence of comparisons that you have to make

those optimal sequences look really pretty random. They do not look like engineered things. Sort of an interesting case where the optimal algorithm is one that you can sort of find by searching the computational universe. But you'd never engineer it. It's just too irregular, too random looking to engineer
in practice. I think the sorting networks the biggest one where we know the optimal network is is 16 elements which is not very big. But that's a case where sort of you can. You can hope that you can find sort of ways to
to kind of
find those kinds of algorithms by sort of automated searching for algorithms. Now, in fact, something I've done quite a bit is kind of searching the space of possible programs for algorithms for doing particular kinds of things. The big surprise is that if you search like. In fact, just last night I was.
I burnt about a thousand hours of of computer time doing a big search for something for a kind of a computational thing for for which you can think of as being algorithms. And I was just earlier today, I had left my computer running all night with, well, a bunch of parallel computers running on this problem and and giving back the results to my computer. And it generated all these pictures. And I was just starting to look through them to try and see whether it found something exciting, so to speak. That's an example of of kind of automated algorithm search.
I think, the
really, the state of the art.
For I have this problem I'm trying to solve with these constraints make me an algorithm that does that.
We don't know a lot about doing that except in the case of sort of very fuzzy things like neural nets. I think that these things that I've done a bunch of of just searching the computational universe of all possibilities and trying to see how that matches up with the things that one is trying to achieve. That's an interesting direction. And I've had quite a bit of success with that over the years. But there's vastly more that can be done in those directions.
Let's see.
Oh.
Let's see.
Well, woman asks, what are some technologies that modern civilization has adopted, developed, and invested so much into that while they might be inferior to alternatives, we're kind of stuck with them because of cultural ossification. I think it's not so much cultural ossification. I think it's that it's just
this sort of economy of technology is such that when you've built this big stack, when you've optimized in this particular direction. It just takes so much effort to go in a different direction. I mean, we could think of a zillion kinds of standards for communication standards. You know, keyboard layouts
ways of representing numbers in computers. There are tons of things like that where just so much is based on them that it's kind of the cost of switching is way too high. And what's the value of switching? Let's talk about keyboard layouts, you know. Qwerty is probably not the best keyboard layout
move and the distribution of words in English and and things like this. There are better ones, but you know, to use a better one. We'd all have to learn how to use that different keyboard, and it could be that in practice the the sort of total time to learn how to use the keyboard, and so on, the the sort of switching cost of having some keyboards in one form, some in another form. It just

would never be worth it. So, even though sort of there's a a better endpoint. There isn't. there isn't a
that there isn't. It's not realistic to get there. Now, it's interesting to compare that with biological evolution. Because I was mentioning earlier. You know, there may be this amazing organism that could be built that could be made with proteins and so on.
But you just can't incrementally get there from organisms that already exist. I mean a classic example of that is wheels where there are microscopic organisms that have, you know, flagella, that rotate. But there aren't macroscopic organisms with wheels instead of legs, and people sometimes argue well, wheels aren't that good because they don't go over rough terrain, and so on. But I bet there are cases where wheels would be, you know, deploy the wheels
for that creature that's trying to run away from its predator and so on.
would be a useful thing. But then there's a question. Well, maybe you just can't get there. Maybe you can't build a wheel with biological materials, because, oh, you can't feed blood supply to a thing that's turning around something like that. There might be a solution to that. Or maybe it's just it was really hard to get to macroscopic size wheels from the biology we have. In other words, there wasn't a path
that took us through organisms that wouldn't be total failures. And the main, the same thing might be you could think of as true in the engineering domain that you can get to a better place. But it goes through things which have such high cost that you'll never go there now, you know, sometimes the
probably it's the case that a different approach really would pay off. I mean there's all these sort of fallacies of sunk costs and things, and so on. Of do you? You know, when do you decide? It's time to pay off the technological debt
and build a new thing. I mean, it's something that comes up for us quite often in software engineering, and so on of when is it? When is it worth just throwing away the old thing, building a new one, or retooling things versus sort of patching the old one?
And for for sure there are. There are
sort of it's. It's a complicated dynamics of the world, and it's a question of what you're optimizing for, whether it makes sense to do that. I mean, I'm trying to think about cases where sort of the the path that's gone down. I mean the fact that you might use, have the incantation https Colon slash slash at the beginning of sort of going to every website. Well, in principle that seems kind of silly in practice. It doesn't really matter, because web browsers just fill it in. Now.
it's just like you could say, well, that's kind of inefficient. You know, how can we deal with things like that? Well, it turns out there's then a layer of technology that comes on top of it like, Oh, you don't want long command names because they're so hard to type.
But then you get autocomplete, and it doesn't matter anymore. So I think you know, papering over these things that are perhaps were not precisely the right initial choice. You know you can. There's a question of whether whether the technology can paper them over, whether whether one should go back and redo it, and I'm sure there are better examples than I'm thinking of right now, of places where sort of redoing it.
you know, becomes more and more difficult. But it really would have been a good idea. It's just we'd have to go back 100 years to do that I mean, I'm kind of thinking that some of the things like laying cables in the ground, and so on, where nobody actually knows where the cables are in City Xyz. You know. It would have been better to put them in some kind of conduit that so you know where all of them are, and so on.

But that wasn't a thing that was being thought about when the 1st electrical cables were being buried and 1st telephone cables were being buried. And now it's easier to sort of have an overlay on that than to go back and fix it.

Let's see,

Hard work is asking what will be the next big thing after AI. So I know what to learn about what will be the next big thing.

Well, I've been saying for a while. I think robotics.

humanoid robotics, and things will be a will be a big thing, but that will be rather driven by AI.

I think the kinds of things that.

let me say the following, there are, there are things where

there is technology that I think is inexorably going to come at some point, but it's super hard to say when. And I think one could be wrong by a hundred years.

so you know. And that's true of lots of technology of modern times. Some of it was portrayed in Science Fiction, you know, half a century or a century ago, and it's just very difficult to know when it will come. I mean, I think a bunch of kind of molecular scale technology and and things. You know, chemistry is about kind of having chemical reactions where molecules sort of randomly bump into each other and will connect, you know, modify each other, or whatever else biology seems to have figured out a lot better how to sort of have this bulk orchestration of how molecules move around to achieve certain objectives, and so on. My guess is that there will be by some method, whether it's in solid state, whether it's not by some method there will be sort of ways to have things happen at a molecular scale that are like computing and under our sort of engineering control. That will eventually be a big thing.

You know, I don't know when I've been waiting for that one for 45 years. And I don't. There's very sort of intermediate technology that one could imagine building to get to that point. But we're not. We're not really there yet.

I mean, there are things.

And when we do get that kind of thing there are an incredible number of sort of interfaces to our biological existence. That will be be very important for for that.

And I think that will allow sort of a bunch of biomedical advances of sort of fixing different things in humans sort of better than the immune system, immune system and patching things up and going and sending in sending in the nanobots to go repair that particular piece of tissue, or whatever else. There are just lots of kinds of things that would happen sort of at the hardware level then, but that hasn't been unlocked yet.

You know I I mentioned well, you know

what tends to unlock new things is some new methodology that suddenly starts to work like Llm. Suddenly started to work. And that's led to lots of new things.

you know, if somebody was able to do. Let's say, Gene sequencing vastly faster, cheaper than it's done right now. So it routinely becomes possible to just, you know, sequence every piece of fish you're about to eat, or something, or sequence your immune system every day to figure out what's happening in your body type thing.

these kinds of things you could imagine, once that technology gets opened up, that a huge torrent of possibilities will will occur as a result of that. And that's usually the thing that some methodology suddenly starts to work. And then there's a torrent of things that happen.

You know. The big methodology that I've been sort of making use of is kind of this computational paradigm. The idea of sort of formalizing things in terms of computation. That's the thing that sort of welcome language is based on. That's the thing I've been sort of trying to do all these years is make this kind of computational notation for talking about the world

and what that computational notation enables is a computational X for all fields X and over time, different fields. X start to suddenly get a computational X, and those are other kinds of that. That's a case where there's an overarching methodology of sort of applying the computational paradigm.

using our computational language, for example, as the as the path to be able to do that, and then being able to go and take those ideas and feed them into this or that field that has not been sort of enabled by that kind of methodology, and those are those are big new things that will happen. Each one of them takes kind of the seeing how to connect kind of this

new paradigm into an existing field.

I mean, it's

that. Just you can. You know, it's something where it's going to be. It's going to require creativity to make that connection.

Once that connection is made, you can then take all the things that have been developed in this computational paradigm and immediately start applying them to this area or that. And that's what I've been doing in the last few years, particularly from our physics project, applying kind of the ideas from the paradigm that emerges from our physics project and applying them to things like the foundations of machine learning foundations of biological evolution. I'm hoping soon foundations of economics.

foundations of neuroscience, etc. These are all things where sort of things are getting opened up. and there are lots of possibilities there. But those are things on the conceptual side. There are also things on the kind of what's hardware possible side. And you know, it's a good question, sort of if you make some particular kind of thing a thousand times faster, cheaper, whatever you know, how does that change the world? And then the question is, Well, can you actually do that? Make the thing, you know faster, cheaper, whatever.

And and that may be something it's very hard to predict some of those things. It's like, it's probably possible. But it's really hard to predict. I mean, you know, maybe it'll be, you know. Maybe nuclear fusion will be cracked. There are, you know. There are lots of. There's a path to do it, but there are lots of practical problems, or maybe cold fusion will turn out to work.

I kind of think there is a way to do that.

and you know it'll be discovered. It isn't, you know, dissolving hydrogen and palladium, which was the original idea that everybody said was nonsense, although I'm not as convinced about that myself. But the

you know, maybe suddenly somebody will discover that if it's just this alloy of palladium that has this characteristic, and you feed the hydrogen in just this way. Those hydrogen atoms will get close enough that they'll start undergoing fusion. Of course. What happens then to the material? Once you've got fusion happening and neutrons spraying out, I don't really know. But in any case, maybe cold fusion will get discovered.

That happens. Then all sorts of things get enabled as sort of a vast thing that gets opened up of of things where sort of there's it's trivial to get power to lots of things where that wasn't possible before.

Let's see.

All right. Maybe one more question. Then I have to go back to my day job.

okay, Latch asks, do you think there'll be in the short to mid midterm, an AI architecture that manages to synthesize mental images to the level. Most humans do visual spatial kinds of patterns. I mean, if you mean by that, will

ais produce images that seem as realistic as the things we see with our eyes for sure that will happen. It's already happened to some extent the mental images that we conjure up.

We have a very hard time knowing what quite what they are.

you know we can describe them. I'm imagining, you know a purple cone or something, or I imagine this thing where this happened. But it's mostly taking a mental image which is somehow formed in our brains and verbalizing it in a very coarse way with words.

So I don't think we know what our mental images are. You could try to use functional MRI, for example, to go and look at activity in one's brain, and try and deduce what images one's thinking about. From that. There are a number of experiments that have been done on that. And there's some enthusiasm these days for brain computer interfaces that could be basically thought to picture type interfaces rather than text to image. You've got thought to image that hasn't worked yet in a serious way. Will it work?

I'm a bit skeptical, actually, because I think that the

when we think of something the unformed thoughts in our minds are unformed, they really don't get to be a thing that's transmittable to anybody or anything else

there, and and that really, it's only once we've got that thought to the level where we can start to verbalize it, that the thought is formed enough that it is communicable. So I think you know, we say we imagine a mental image. But but

we, you know, I don't know. Maybe I'm just bad at doing mental imagery. I don't know. It's you know. There are people who have very different levels of performance and things like being able to do 3D. Mental rotation given. That you have. One of my kids, for example, is very good at this. Given that you have an object, you know, and it has this shape. What happens if you were to rotate it, what would it look like? And so on. I can't do that at all. So. But some people can do that well.

And I think and perhaps so, perhaps I'm I'm

impoverished in terms of mental images. I mean one of the one of the sad things about the field of mental imagery in neuroscience is one of the people who is a prominent researcher in that area.

Also, just didn't form mental images really, and came to all kinds of conclusions in that field based on that fact, even though other people have a different experience. I mean, just like other. Some people have, you know, synesthesia. Where, given a letter, you think of a color, and so on. And you know again, one of my kids. A different kid has that. And I've always been meaning to go back and try and see whether

the sort of early early learning of letters happened to be on some, you know, board that had, you know, an A in green and a B in purple or something, and whether that's how the synesthesia got developed. I have no idea whether it's really a brain connection type thing of this shape is sort of close to the this set of sort of things that are perceived from color receptors, and so on.

But I I this whole question of of sort of can we emulate the mental image? I think the only thing we really know is, can we make an image that

when transmitted through our eyes we'll have the sort of sensation of a certain kind of image, and that we're well on the way to doing. If we ask about brain computer interfaces where we're picking things up from the innards of our brains and turning those things into sort of visible eye. Seeable images. That's an interesting possibility. If we imagine sort of the other way around, of sort of of of of sort of stimulating our brains to see a particular image. I think that's far away at this point.

very far away. I think you know the moment when the you know the deep brain stimulation, you know, array of electrodes is going to cause us

to imagine this elaborate scene. I don't think that's near at hand. I mean, of course, one knows that that in I don't know neurosurgery and things when you, when you sort of excite a particular spot in the brain. You can end up having particular verbal responses to that.

So, but I think that's a little bit of a different thing. I think that's something where you're kind of stimulating. I don't really understand that phenomenon. I don't know to what extent it's a reproducible phenomenon, and you know it's very difficult to do reproducible experiments on brains, because they learn. I mean, if you say it's like, you know you, it's

you know you go

to, you know, an optometrist or something, and they say, Can you, can you read off these letters? Which line can you read? And I certainly remember in the distant past, just saying, You know, they say, Wow! You can read so so many lines, and that it's like it's the same chart. I just saw it with a different lens. I have a good memory. I remember these letters, you know. I learned it so it isn't a good test anymore. And but so you know those experiments are difficult to do.

All right. I think I should.

wrap up. I'm sorry I have to. There's 1 last question here from Weasel. Do you think there will ever be a decent food? 3D printer in the future as promised? I think the chocolate was the only thing that ended up being viable.

You know I've seen these printers, these sugar printers and I. I seem to remember one event we did. We had some 3D. Printed chocolate thingies. But yeah, I haven't seen that. I have to say it's it's 1 would think that the sort of the very High End restaurant or something would start to have 3D. Printed food or that one would have sort of

for for our company. We have this 3D logo. This spiky object, the rhombic hexacontahedron, and it's very 3D. Printable and making sort of 3D. Printed

sort of

food food stuff in that shape will be seem like a natural thing to do. I think.

I think it. Yeah. I haven't seen that. I don't know what the issues are there. I think 3D. Printing in general has sort of segwayed into this sort of professional 3D printing of printing things for medical purposes, you know, teeth, bones, things like this, or printing things that are sort of weird shaped blades for turbines, or something like that. There's sort of the professional 3D. Printing world, which has very much taken off.

but the more. And then there's the very amateur. It's just a cute thing to be able to print this thing out of rather crummy plastic. 3D. Printing world. The intermediate scale of that seems to have more or less vanished. I mean, I needed to 3D. Print something. What? A year or so ago!

And it was fairly elaborate thing. And I had a very hard time with that. I was trying to print something kind of complicated. It was in multiple colors, etc, etc, etc. And all these companies that used to exist that did that have disappeared. And it's like, Well, do you want a production run of 10,000 of those things? Not? Oh, we can just run off this 3D printout

you know I do have to say in terms of making shapes of things. I'm going to tell 2 stories, and then I really have to wrap up. One was an early

sort of conference about new kind of science from 20 years ago, or something. Now, where we had had a cake made in a 3D solar automaton shape, with, I think, that was not explicitly 3D printing that was more just hand assembly and things. The problem was

the the sort of cake engineering had not been done adequately, and it was a very sad, sorry thing, because this very lovely pyramidal cake just sort of crumbled before one's eyes when when. So that was one sort of shapes, a hard type thing. I have to tell another story
from the beginning of the 19 nineties when we were making a trade show booth for our company, and we wanted to kind of fly a balloon up above the the trade show booth.
and we had contacted some company. I mean, I wasn't the frontline person doing this, but actually, the person who was the frontline person was a fairly sophisticated mathematical person. We'd contacted this company, and we really wanted to make a a knot, some truffle knot, or something in the as as the shape that we would have as our balloon, you know, a helium balloon above our booth if the trade show allowed such a thing.
And so this company contacted them, and they're like, yes, we can make a balloon in any shape. We've made chickens we've made, you know, bananas, we've made any shape, and then
we sent them this truffle knot, and they're like very, very confused because, in fact, what they meant was any shape that is sort of continuously deformable from a sphere. But that was that level of sort of mathematical sort of thinking about shapes
probably was something that absolutely never come up in their business. But it's just one of those stories of, you know. We can make a balloon in any shape. We can 3D. Print food in any shape, so to speak. Well, there end up being these practical problems about what you can do. But yeah, I wonder what's happened to the food printers. I think you know the fundamental problem of 3D printing at some level.
is it? It's still kind of messy, you know. You, you build things and you get a lot of little granules there, and you have to brush them off, and you you have to clean things and so on. My guess is with things that are that are edible.
that that makes you know it all gets very complicated. I mean, we know, you know, people who decorate cakes have something, you know, have a nozzle that they're squirting, icing out of, and that's not that different from a 3D printer nozzle. But my guess is that if you try and do that on a large scale, the thing gets very messy, and I'm sure if you're a cake decorator, you kind of know you can do it for a little while, and you have to clean the thing and so on.
and probably for a 3D. Printer. I don't know how that works, I mean, one could imagine the printer would would squirt, you know, whatever it is, sugar through the nozzle for a while, and then we'll go into some kind of cleaning cycle and go back to doing it. I don't know why that hasn't happened more. A good question. I should check it out
alright. Well, thanks for lots of interesting questions. There was fun to try and answer these with you guys. And I
look forward to chatting with you another time.
Bye, for now.