[Transcript] The Ezra Klein Show / A.I. Could Solve Some of Humanity's Hardest Problems. It Already Has.

From New York Times Opinion, this is the Ezra Klein Show.
So I think you can date this era in artificial intelligence back to the launch of chat GPT.
And what is weird if you talk to artificial intelligence people about that is they'll tell you chat GPT.
It was just a wrapper, an interface system.
The underlying system GPT-3 had been around for a while.
I mean, I'd had access to GPT-3 for quite a while before chat GPT came around.
What chat GPT did was it allowed you to talk to GPT-3 like you were a human and it was a human.
So it made AI more human.
It made it more able to communicate back and forth with us by doing a better job mimicking us and understanding us,
which is amazing.
I don't mean to take anything away from it, but it created this huge land rush for AIs that functionally mimic human beings.
AIs that relate as if they are human beings and try to fool us into thinking that they're human.
But I've always been more interested in more inhuman AI systems.
When you ask somebody who's working on artificial intelligence, including people who believe it could do terrible harm to the world,
why are you doing it? What's the point of this?
They don't say, oh, you know, we should risk these terrible consequences because it's fun to chat with chat GPT.
They say, oh, AI, it's going to solve all these terrible scientific problems we have, clean energy and drug discovery.
And, you know, it's going to create an era of innovation like nothing humanity's ever experienced.
There aren't that many examples, though, of AI doing that yet.
But there is one, which you may have heard me mention before.
And that's AlphaFold, the system built by DeepMind that solved the protein folding problem.
And the protein folding problem is that there are hundreds of millions of proteins.
The way they function has to do with their 3D structure.
But even though it's fairly straightforward to figure out their amino acid sequence,
it's very hard to predict how they will be structured based on that.
We were never able to do it.
We were doing it one by one, studying each one for years to try to figure out and basically map it.
And then they build the system, AlphaFold, which solves a problem,
is able to predict the structure of hundreds of millions of proteins, a huge scientific advance.
So how did they build that?
And what could other systems like that look like?
What is this other path for AI?
This more scientific path where you're tuning these systems to solve scientific problems,
not to communicate with us, but to do what we truly cannot do.
Demis Asabis is the founder of DeepMind.
DeepMind is owned by Google and recently Asabis was put in charge of all Google AI.
So now it's called Google DeepMind and he runs all of it.
That makes him one of the most important people in the world,

charting the future of artificial intelligence.
So I asked him to come on the show to talk me through the development of AlphaFold,
how it was built, what came before it, what could come after it,
and what that says about the different ways and the different pathways forward for artificial intelligence.
As always, my email is reclineshow at nytimes.com.
Demis Asabis, welcome to the show.
Thanks for having me.
So tell me about the founding of DeepMind.
You were pretty young at that point.
How did you decide to start it?
What was the vision for it?
Well, actually, my background in AI starts from way before DeepMind.
So I started actually in the games industry, writing computer games and some best selling games, actually games like Theme Park and Black and White and other games that I worked on in my teenage years.
And they all had AI as a core component of the game.
So let's take Theme Park.
It was a simulation game, came out in 1994.
And you basically created a theme park and lots of little people came in, played on the rides and bought stuff from your stalls.
So there's a whole kind of simulation underlying it.
And AI was the core part of the gameplay.
So I've been working on AI for nearly 30 years now in various guises.
So if you look at my career, I've done lots of different things,
but they were all doing something like an effort like DeepMind working on AGI.
And so all the things I did, the neuroscience, the computer science, undergrad, PhD in neuroscience, was all for gathering information and inspiration for eventually what would become DeepMind.
So I played Theme Park back in the day.
And if you ask me now and you say, hey, Theme Park, that game you played in the 90s, was that AI?
I would say no.
There's a classic line to paraphrase that AI is anything the computer can't do yet.
But now you look back, you think, oh, no, that's just a little video game.
So when you say that was AI, what are you saying by that?
And to you, what is artificial intelligence?
And what's just machine learning or statistics or some less impressive function?
Yeah, so back in the 90s and with games like Theme Park at the time, that was pretty cutting edge.
We were using relatively simple techniques, Elio Automata and relatively narrow AI systems,
sort of logic systems really, which was in vogue back in the 80s and 90s.
But it was AI in terms of making a machine do something smart and actually adapt to,
automatically adapt to the way the gamer in that case was playing the game.
So the cool thing about Theme Park was that, and why it was so successful and sold millions of copies around the world,

was that everybody who played it got a unique experience.
Because the game adapted to how you were playing.
It's very primitive by today's standards of learning systems.
But back then, it was pretty groundbreaking.
It was one of the first games along with SimCity to do these kinds of complicated simulations under the hood powered by AI.
As to your question about what is AI, I think AI is the science of making machines smart.
And then a sub branch of AI is machine learning, which is the kinds of AI systems that learn for themselves
and learn directly from data and experience.
And that's really what, of course, has powered the renaissance of AI in the last 10, 15 years.
And that's the sort of AI that we work on today.
Can you just spend another second on that?
You mentioned a minute ago that what was happening in Theme Park was logic based systems, which I think is at least one of the other branches.
Can you describe the difference between some of the more logic based or rules based or knowledge encoded systems
and deep learning and some of the other things that are more dominant now?
Sure.
So if you think of AI as being this overarching field of making machines smart,
then they're broadly speaking two approaches to doing that.
One is the classical approach and the sort of approach that was done for the first few decades of AI research
since the 1950s was logic systems.
So this is the idea of the programmers or the creators of the system.
They effectively solve the problem, be that playing chess or controlling little characters in a game.
And then they would program up these routines and heuristics,
and then effectively the system would then deal with new inputs and execute those heuristics.
And you can beat pretty powerful systems.
They're sometimes called expert systems.
And the most famous of that is probably Deep Blue IBM's chess program.
They beat Gary Kasparov, the world chess champion at the time in the 90s very famously.
And that was probably the pinnacle of expert systems.
But the problem with them is that they're very brittle and they can't deal with the unexpected, of course,
because they can only do what the programmers have already figured out and the heuristics they've already been given.
They can't learn anything new.
So machine learning is the other approach to solving AI.
And it's turned out to be a lot more powerful and a lot more scalable,
which is what we bet on as well as when we started DeepMind.
And that's the idea of machines, the systems learning for themselves,
learning the structure, learning of heuristics and rules that they should do for themselves,

directly from data or directly from experience.
The big contrast for our first sort of big famous program was AlphaGo,
which learned how to play the complex game of go for itself
and actually came up with better strategies and heuristics than we could have ever thought of as the human designers.
We're going to get to AlphaGo, but I want to hold for a minute in the founding of DeepMind.
So you've been a game designer.
You become a neuroscientist and you do actually some important peer-reviewed research
and highly cited research in that field.
That's a big jump. You got to work pretty hard to become a neuroscientist is my understanding.
So you're an academia, you're doing this research, it's going well as best I can tell,
and you start deciding you're going to found this AI company.
So take me in that moment of decision making.
Yeah. So again, if you have in mind that I was planning from, I guess I was around 15, 16,
when I was doing this programming on these computer games and theme park and sort of 17 years old,
that's already when I decided my life was going to be and my career was going to be about AI
and making AI happen.
So all the other things I chose were in service of that, including the PhD.
So I did my undergrad in computer science.
I got trained in coding, engineering and mathematical side of computer science,
which I love the theoretical side, Turing machines, all of these types of things, computation theory.
And then I decided for my PhD after I ran my own games company for a while,
I went back to academia to do a PhD in neuroscience.
And I chose cognitive neuroscience because first of all, I've always been fascinated by the brain.
The brain is the only existence proof we have in the universe that general intelligence is possible.
So it seems worthy of studying that specific data point.
It's probably not the only way that intelligence could come about,
but it's certainly the only way that we're aware of and that we can study.
And of course, it's fascinating subjects in itself.
But the reason I did the PhD is I wanted to learn about the brain,
maybe get inspiration for some algorithmic ideas, architectural ideas.
And indeed, that's what did happen in things like memory replay, reinforcement learning and things like this,
that we then used in our AI systems.
And also learn how to do cutting edge research too,
and actually learn how to use the scientific method properly and things like control studies and so on.
Really, you learn all of those practical skills by doing a PhD.
You said something about founding DeepMind that I've always found striking,
which is, quote, I want to understand the big questions, the really big ones
that you normally go into philosophy or physics if you're interested in.
I thought building AI would be the fastest route to answer some of those questions.

And so a lot of people might want to know about the nature of the universe.
The idea is I'll go work on it directly.
I'll get a physics PhD, a mathematics PhD, a PhD in chemistry or neuroscience,
and I'll go solve those problems.
And your theory of the case was I will get this training and build something else to solve those problems.
Why do you think that intermediary intelligence is necessary?
Why not just do it yourself?
It's actually a great question.
I mean, I've always been fascinated by the biggest questions.
I'm not quite sure why that is,
but I've always been interested in the nature of the universe, nature of reality, consciousness,
the meaning of life, all of these big questions.
That's what I wanted to spend my life working on.
And indeed, physics was my favorite subject at school.
I love physics.
I still love physics.
I still sort of try to keep tabs on interesting areas of physics like quantum mechanics and so on.
But what happened is a lot of my scientific heroes in my early years were physicists,
so Richard Feynman and Stephen Weinberg, these kinds of people.
But I actually read this book.
It must have been in high school called Dreams of a Final Theory by Stephen Weinberg.
And it's his book, and you know, Nobel Prize winner, amazing physicist of his work
on trying to come up with a unified theory of physics, you know, to unify everything together.
And I remember reading this book and I was very inspiring book,
but I remember concluding, wow, they hadn't actually made that much progress.
And these incredible people, you know, that I had really admired, you know,
you can think of all the physicists since post-World War II, right?
These incredible people like Feynman and Weinberg and so on.
And I just remember thinking, gosh, I wasn't that convinced they'd got that far.
And then I was thinking, even in the best-case scenario that you might be able to follow their footsteps,
which is a big if, given how brilliant they were,
you still might not be able to make much progress on it in the whole lifetime of, you know,
an amazing career like they had.
And so then I started thinking, well, perhaps the problem is,
is that we need a little bit of help and a little bit of extra intellectual horsepower.
And where can we get that from?
Well, of course, I was also simultaneously programming and doing games
and falling in love with sort of computing.
And that was my real passion.
And I just realized that working on AI, I could satisfy both things at once.
So first of all, it would open up a door to insights into what intelligence is

and how the brain works and, you know, as a comparator and so on,
when those are some of the biggest questions already.
But it could also help with science and physics and help experts in those areas
crack the problems and the questions they care about.
So it seemed to be like the perfect sort of meta solution in a way.
And when I made this realization sometime in high school, that's when I decided, you know,
I was going to work on AI and not go directly, say, for physics
and try and build the ultimate scientific tool.
So let's talk about how some of this experience comes together for you
because one of the early contributions of DeepMind
is you're trying to build this new kind of intelligence
and the way you're trying to build it, train it, test it is on games.
And now it feels like a weird thing to hear,
but it takes you a very long time to build a system
that can score even a single point in Pong.
So tell me first about the decision to begin training AIs on video games
because that doesn't seem totally intuitive.
You want to answer the fundamental questions of physics
and you're building something to play Pong?
It seems ridiculous. Why?
Yes. That was one of the early decisions, I think, that served us very well
when we started DeepMind was a lot of people who were working on AI at that time
were mostly working on things like robotics and sort of embodied intelligence
and that's an important branch of AI, of course, and places like MIT
where I was doing my postdoc, that was a real bastion of that type of work.
But what I realized was that the researchers ended up spending most of their time
fiddling around with the hardware, you know, and the server motors
and they'd always break and the robots are expensive and they're complicated
and they're slow and I sort of realized that it would be better to work in simulation
and we could, you know, run millions of experiments in the cloud all at once
and get much faster learning rates relatively cheaply and quickly
and simply if we were to do that.
The other advantage of games, of course, is that they've been built to be challenging to humans
and you can use top human players as a great benchmark.
The other thing is they have clear objectives, right?
You to win the game or to maximize the score
and those kinds of objectives are very useful if you want to train reinforcement learning systems
which we specialize in that are reward seeking and goal directed, you know,
so they need an objective to solve.
So games are fantastic for all of those reasons.
And the other cool thing about games is, of course, you can go up the ladder of complexity of games
by just going through the different eras of games.
We started with the 1970s with the simplest Atari games like you mentioned, like Pong

and then eventually we went all the way up over, you know, almost a decade of work
to the most complex modern computer games like StarCraft.
And so you can keep on increasing the difficulty of the test for your AI systems
as your AI systems are getting more sophisticated.
So, you know, it turned out to be a really efficient way to proving ground really
to test out these algorithmic ideas.
So I don't want to skip over too quickly what's happening here.
I think when I say the sentence, oh, you built an AI that can play the game Pong,
that sounds simple, but it takes you a long time to do that.
It takes you a long time to score any points in Pong.
And then you begin to see exponential dominance of the Pong game,
which is an interesting dynamic here that I want to talk about too.
But tell me what you're actually doing there.
Sure. So the key thing here is, and this is the difference between what I was doing in the 90s
with my early games career where we were directly programming these expert systems.
Of course, you could do that easily for something like Pong.
But what the key breakthrough that we had that sort of underpinned this whole new field really,
which we call deep reinforcement learning, combining neural networks with the reward-seeking
algorithms
is that we played these games directly from the pixels.
So all we gave is the input to our systems.
The first system we built was called DQN to play these Atari games.
What did DQN stand for?
DQ network.
I was hoping that was going to be more fun.
No, no, it was a very technical name.
We gave that one at the beginning before we got better at naming things.
It just refers to the technique that we used.
And so the big innovation and the big breakthrough was to use just the raw inputs,
in this case the pixels on the screen, 30,000 pixels roughly on the screen,
and not tell the system anything about the game.
Not what it was controlling, not what would get score, not how it loses points or loses a life.
Any of those things, it had to figure out for itself from first principles by playing the game,
just like a human would learn.
That was the key thing.
So it learned for itself, and the second thing that was key was building these general systems.
And that's what the general is in AGI, and I'm sure we'll come back to that,
is that one single system that can play any of the Atari games
and the same system out of the box can play all of them to sort of superhuman level,
like world record scores.
And so those two elements, the generality and the learning, are the key differences.
So I want to spend just a moment here on this expert system versus deep learning.
So if you've been building an expert system, a logic system,

you're trying to tell the system how Pong works.
It's like, okay, there are these two paddles, and there's a ball,
and what you want to do is get points,
and you're basically trying to break the game of Pong down into rules,
encode the rules into the system,
and you figure out if you've found all the rules by how well the system does
versus deep learning, where are there any rules there?
Are you just telling the system, if you get a point that's good,
experiment until you do,
and it gets basically the digital equivalent of a doggy treat every time it does well
and then tries to do more of that well.
That's right.
So that's exactly the difference.
So expert systems, you as the human programmers and designers are trying to break down
this complex problem, be that playing chess or playing an Atari game,
into the set of heuristics and rules.
So sometimes there are rules, but they could be probabilistic,
so there could just be heuristics,
and that's what the system then uses to try and make decisions.
So it's effectively using what it's been given.
But then with machine learning, using techniques like deep learning,
type of machine learning or reinforcement learning,
which is the doggy treat thing that you mentioned.
So you get a point, so it gets a reward,
and then it's more likely to do those actions again.
Those things learn for themselves.
So you just give it the high-level objective.
You know, you say win a game, get a certain number of points,
and then it has to figure it out effectively what those heuristics are for itself.
And so we're going to talk more about this,
but it's in this divergence where, as I understand it,
the question of alignment problems really begins to creep into the industry.
Because if I'm encoding all these rules,
if I'm encoding my understanding into a computer, then I might miss things,
but the computer is basically running off of what I've told it to do.
Whereas if I've told it to get points,
and it doesn't even know what it's doing except for getting points,
I mean, as you say, it's learning from pixels.
It doesn't have an understanding of the game Pong,
and ultimately it's winning points,
but it still may not know it's playing Pong, right?
I've never told it it's playing Pong.
It's not working with that kind of generalized sense of the situation.

It might get points, but if it can figure out another way to get points,
it'll do that too.
Let's talk a bit about what it means in terms of
what the system is doing versus what we often think it's doing
that comes from when it is doing the learning itself.
You know, this is a very interesting question of when you do the learning itself,
you can guide that learning by giving it certain types of data to learn from.
You can set the high level objectives.
You could even set sub-objectives as well to kind of guide it on the way.
How do you win a game?
Well, you've got to get a certain number of points or something like that.
So you can sort of give it some sub-objectives,
or it can discover that for itself.
But really, you're sort of more coaxing the system
rather than with the logic systems where you're directly programming that in.
So you have a little bit less direct control over that.
And that, of course, is then linked to what you mentioned,
the alignment problem, which is sometimes you may care
not just about the overall outcome,
you may care about how it got there, right?
And if you do, then it matters the way that it solves it.
And so then you're kind of leaving the solution
or the type of solution or the approach up to the system itself.
And what you're caring about is the objective at the end.
So if you care about the approach too,
then you have to add extra constraints into the system
or give it some feedback about the approach too,
which is this sort of reinforcement learning with human feedback
that was now in vogue where you can get that feedback directly from human ratings
or you can do it another way.
You can give it sort of further sub-objectives that help it guide it
as to what type of solution you want.
Well, we've talked a bit here about deep learning,
but can you describe reinforcement learning?
What is it? How does it differ from deep learning?
How do they work together?
So both deep learning and reinforcement learning
are types of machine learning,
and they're very complementary.
And we specialize in both and have done from the start of DeepMind.
So deep learning is on your hierarchical neural networks,
really complex stacks of neural networks,
very loosely modeled on brain neural networks.

And the objective of those neural networks
is to learn the statistics of the environment that they find themselves in
or the data stream that they're given.
So you can think of the neural network,
the deep learning as building a model of the situation.
And then the reinforcement learning part
is the bit which does the planning and the reward learning.
So effectively it's kind of like a reward-seeking system
that is trying to solve an objective that you give it.
So for example, we'll be in a game trying to maximize the score.
So for every point it gets, it's sort of like a little reward.
Animals, including humans, learn with reinforcement learning.
It's one of the types of learnings we do.
And you see that really well with, as you mentioned earlier, with dogs,
and you give them a treat if they do something that you like
or they're well-behaved.
And then they're more likely to do that again in future.
And that's exactly very similar in a digital form
with these reinforcement learning systems that we build.
Effectively they get treats, a reward,
when they achieve a certain sub-objective.
And so the cool thing is, is that you can combine these two types of systems together,
and that's called deep reinforcement learning,
where you have the model,
and then you also have this goal-seeking or reward-seeking planning system
on top that uses that model to reach its objectives.
So I want to give ahead a bit now to the system that most people,
at least you should know DeepMind for,
and that's AlphaGo.
So tell me what AlphaGo is,
how it differs from the Pong system we're talking about here,
why Go was an important benchmark or milestone to try to topple?
Yeah, so in effect, AlphaGo was the extension of the work we'd done in Atari,
but sort of the pinnacle really of what you could achieve in games AI.
So Deep Blue, as I said earlier, beat Gary Kaspar for chess in the 90s.
That was one big first pinnacle in games AI,
but the next sort of Mount Everest, if you like, was beating the world champion at Go.
And you couldn't use expert systems.
You had to use these learning systems,
because we as even the best human Go players,
they don't understand the game well enough to break it down into these heuristics and sub-problems.
It's also partly to do with the nature of Go as a very aesthetic game,

it's a very intuitive game with a lot of patterns.
So it's quite different from chess, which is a more of a calculation game.
Even top Go players will tell you why did they play a certain move,
they'll say it just felt right.
So they're kind of using their intuition,
which is not something that we often associate with computer programs,
being able to do something that mimics intuition.
And so with AlphaGo, what we did is we built a neural network that modeled the game of Go,
figured out what were good moves in certain positions,
who was likely to win from a certain position, the probability of either side winning,
and that's what the neural network was predicting.
And then on top of that, we overlaid this reinforcement learning system
that would make plans, do Monte Carlo tree search,
using the model to guide its search so that it didn't have to search millions and millions of moves
and it would be intractable because it goes too complicated to search everything,
you have to narrow down your search.
And so it used the model to search the most fruitful paths
and then try and find the best move that would most likely get it to a winning position.
And so it was a kind of culmination of five, six years' worth of work
in our work in deep reinforcement learning.
What I always found striking about that story is so you beat one of the Go World champions,
it's this big moment, and then shortly thereafter you beat yourself,
which is to say you have AlphaGo, which is the system that beats release at all,
and then you create AlphaZero, a system that just stomps AlphaGo.
So what was the difference between AlphaGo and AlphaZero,
and what was to be learned there?
So AlphaGo was a pretty general system in that it learned for itself
the motifs and the strategies around Go,
and in fact it created new strategies very famously in that in the World Championship match
that had never been seen before, even though we've played Go for thousands of years now,
to a couple of thousand years.
So that was pretty incredible to see.
The first version of AlphaGo actually was bootstrapped by looking at all human games
that had been played on the Internet, and there's a lot of Internet Go servers
in very popular in Korea and Japan and China.
So it's using human games as training data.
It used human data as training data.
It also had specific things, knowledge about Go encoded in it
to do with the symmetry of the board and some other things that were specific to Go.
So it was a pretty general system, but it was specialized around Go data,
the human data that it learned from, and also some specific things about Go.
Now, once we beat the World Champion, Lisa Doll,
we then, this is quite common for us is once we do that,

we always try and look back at our systems, in this case AlphaGo,
and we try and remove anything that was specific to that task to make it more and more general.
And so that's what AlphaZero was.
AlphaZero was the next version, and that is able to play any two player game.
So it doesn't matter whether it's Go or chess or backgammon,
any game you can put it in there, or Japanese chess,
so-called Shogi, any two player game.
And it doesn't need any human data either, because what it does,
it starts off completely random.
So imagine a blank slate neural network doesn't really know anything about the game or strategies.
And what it does is it plays itself millions and millions of times,
different versions of itself, and it learns from its own data and its own experience,
and then eventually it becomes actually stronger than these individual programs like AlphaGo
that were trained on human data.
And it doesn't require any human data.
It starts literally from random and explores the space of Go or chess for itself.
And of course, it means it also comes up with incredible new strategies,
not constrained by what humans have played in the past,
because it doesn't have any knowledge of that.
One of the things I think about, and that we will talk about later in this conversation,
is the question of how smart something that is working with our data
and working with the world as we understand it can really become, if that tops out somewhere.
And one version of saying not that smart might be to say,
well, it's kind of constrained by what we know.
It's got to work with what we know.
And if we haven't done that much on something, well, it can't do that much on something in most cases.
But this is a case where, with some very basic rules,
it actually turned out that it was being held back by what we knew,
that because human beings are prone to fattishness,
because we follow in the footsteps of those who came before us,
because we're taught by others, and they tell us, no, that would be a crazy move.
You know, everybody who's done it before did it this way.
I mean, that does help us get better, right?
Cultural learning and evolution is the core of our species advancement.
But it also turns out that that means huge swaths of useful strategy,
information, ideas have been cut off the board because we just don't do that.
And so in terms of reasons to think these systems could really be remarkable from our perspective,
the fact that it was being encumbered by everything we knew about Go,
as opposed to launched forward by everything we knew about Go,
to maybe put less sentiment on the word here just strikes me as profound.
Yeah, look, it's an interesting take.
I mean, of course, this is what happens with our cultural civilization,

is that we do get into local maximas, one could say, in something relatively prescribed like a game.
In fact, I talked to the Go experts after AlphaGo made these new strategies,
most famously Move 37 on Game 2 of the World Championship match,
was this astounding new move.
And I asked all the Go experts afterwards about that move,
and they said, we're just told that that's a bad move to make in that early in the game,
but they can't really explain why.
They just said their teachers would just basically shout at them.
So it's interesting that that is a cultural norm, then,
that actually limits our creativity and expiration.
So what I'm hoping is these systems will expand our own minds,
and I think this is actually what's happening Go and in chess,
again, in very prescribed areas,
but where people have started being more creative themselves.
Oh, actually, we don't.
Maybe we should question some of those cultural norms,
and perhaps we would get further.
I believe that's what has happened in Go and other things.
Now, the cool thing is coming back to using AI systems for science.
If we talk about that, then science is the pursuit of new knowledge and understanding.
And so you can now see, I think, of course, in games, it's just games,
and they're fun, and I love games, to death,
and my huge passion of mine, but in the end, it's just a game.
But for science, you're actually discovering a medicine,
you're discovering important new knowledge.
There's no reason to assume that isn't going on in another cultural activity,
in this case, science.
And I think these tools could, in themselves,
help us discover new area regions of knowledge,
but also inspire the human experts to explore more as well in tandem.
Let's shift into science, because this is where DeepMind begins creating something
that isn't just winning games, but is actually creating an advance.
And I've said before on this show many times that,
of all the AI systems that have been released,
the one I've always been most impressed by and interested in is AlphaFold,
which is a DeepMind system.
So tell me what AlphaFold is, what the problem is,
how you come to decide that that's something that your systems can take on.
I mean, you're doing games, and then you move to this.
What is AlphaFold?
So we were doing games as the testing ground,
but we always had in mind, and I always had in mind the very thing I was thinking about
as a teenager of using AI as a tool for science.

And once we'd mastered a lot of games,
the idea was that these systems would be powerful enough and sophisticated enough
we could turn them onto very important real world problems
and real world challenges, especially in the sciences.
So AlphaFold, we pretty much started the day after we got back from Korea in 2016
and the Lisa Dolmatch, and that was our next big grand challenge.
And AlphaFold is our program to try and solve the problem of protein folding,
as it's known.
Proteins are the workhorses of biology.
Basically, every biological function in the body is mediated by proteins.
Proteins are described by their amino acid sequence,
so you can think of it loosely as the genetic sequence for a protein,
and that's a kind of one-dimensional string of letters you can think of.
But in the body, they scrunch up into a 3D shape very, very quickly.
And it's the 3D shape of the protein that governs its function, what it does in the body.
And so the protein folding problem, in essence, is can you predict the 3D shape of the protein
directly from the amino acid sequence?
The reason it's so important is that a lot of disease is caused by proteins misfolding or folding wrong,
and also if you want to design drugs to combat diseases,
you need to know what the surface of the protein, therefore the shape of the protein is,
so you know which parts of the protein to target with your drug compound.
So it's hugely important for many, many biological research questions.
To just give an example of this, people may have heard over time that coronavirus is a spike protein,
and that's not just an aesthetic point.
The fact that it has this somewhat spiked folding structure is crucial to the way it actually works.
Do you want to just maybe use the coronavirus protein as an example of what you're saying?
Yeah, so that's a great example, and we worked on that as well with the alfalfold system.
So yes, the spike protein is a thing in a sense that sticks out of the virus.
So that's what you want to latch onto with a vaccine or a drug to kind of block its function,
so it doesn't attach to the body or the body's cells in a certain sort of way.
So it's the protein structures that do all the mechanics of that.
So if you understand what the protein structure looks like, that spike looks like, the shape of it,
you can design something that fits like a glove around it to block its action.
So that's a great example of the criticality of protein structure.
And what made you think that protein folding is like games?
What is the analogy you're drawing here?
When you say, I come back from doing go and I decided to work on protein folding,
what are you seeing here?
Because I would not naturally see a connection between those two questions.
No, it seems quite far apart, but actually depends on if you step back and look at a sort of meta level,
they have a lot of things in common.
And by the way, protein folding is one of a number of scientific problems,
big sort of grand challenges that I came across in my career.

Actually, protein folding, I came across in the 90s in my undergrads.
I had a lot of biologist friends who were obsessed about this at Cambridge
and actually went on to do their whole careers on protein structures.
And they explained to me the problem and I thought it was fascinating.
And I also thought it was a perfect problem for AI one day to help with.
So I kind of filed it away and then 20 years later,
brought it out of the filing system in a sense and decided that that was the first big grand challenge we would apply our learning systems to.
The reason I think it's similar that sort of later on, actually, while I was doing my postdoc,
the second time I came across protein folding was in the late 2000s
where there was this game called Fold It.
It's called a citizen science game.
You may have come across it.
So basically, a lab had created a game, a puzzle game,
where involved people folding proteins in a three-dimensional interface.
I don't think it was a very fun game,
but they made it into this sort of quite user-friendly interface.
And a few tens of thousands of amateur games players got quite obsessed with it.
It got released, I think, in 2008, 2009.
And I remember looking into this and thinking,
wow, this is pretty fascinating if you can get people to do science by playing a game.
That seems like a great idea.
So my game's designer part of me was fascinated by it.
And so what happened when I looked into it is that some of these gamers,
who, by the way, a lot of them knew nothing about biology, right?
They were just gamers.
They'd figured out, presumably, with their pattern matching of their brain, their intuition,
that certain counterintuitive folds of this string of amino acid sequences,
you know, the backbone of the protein,
led it to the right kind of 3D structure.
And they're counterintuitive in that if you just do the fold that gets you locally
to the lowest energy state, which is a kind of 3D search strategy,
you end up with the wrong protein fold.
So you can't do that.
So sometimes you have to do local moves, local bends of the protein,
that actually make the energy landscape worse,
effectively the efficiency of the protein structure worse.
And then eventually you resolve that.
And I remember thinking, combining that with what we then did with AlphaGo,
where in AlphaGo, what had we done?
Well, what we'd managed with AlphaGo and achieved with AlphaGo
is we've managed to mimic the intuition of these incredible Go masters.
So I was thinking, wow, if that was the case with these Go professionals,

you spend their whole life on it.
And then these amateur gamers who didn't know anything about biology
were able to, in a couple of cases, fold a couple of proteins correctly,
then why wouldn't we be able to mimic whatever was going on in that?
Those amateur gamers' intuition.
So that gave me some hope, additional hope, that this would be possible somehow.
And of course, this idea of a game, protein folding being a puzzle game,
was pretty interesting as an analogy as well.
So for several reasons, and also the fact it was a ground challenge,
and it had so many downstream implications and impact, if we were to solve it,
all those factors sort of came together for me to choose that as the next project.
That brings up, I think, another important part of all this.
So when you think about how are you rewarding the system,
how are you reinforcing the system in Go,
well, Go has rules, and you know how you score points, and you know how you win a game.
But when you're predicting the structure of a Hereto4-unpredicted protein,
how do you know if you're right?
How did the gamers know if they were right?
How does the system know if it is right?
What are you reinforcing against?
That was one of the hardest things, and often that's one of the hardest things
with learning systems and machine learning systems,
is actually formulating the right objectives.
I think of it as asking the system the right question.
How do you formulate what you want in terms of a simple to optimize objective function?
And you're absolutely right, in the real world you don't have simple things like scores
or winning conditions, that's obviously in games.
But with proteins and biology, a lot of the cases there are good proxies for it,
like minimizing the energy in the system.
Most natural systems try to be energy efficient,
so you can sort of follow a gradient of the energy gradient,
or the free energy in the system, and try and minimize that.
So that's one thing.
The other thing is protein folding, there is a whole history,
50-year history, or more actually, of painstaking experimental work.
The rule of thumb is it takes one whole PhD, the whole PhD time,
one PhD student and their entire PhD, for five years to crystallize one protein
and then using x-ray crystallography or electron microscopes,
complicated pieces of very expensive, complicated pieces of equipment,
to basically image these incredibly small, complex structures.
It's unbelievably painstaking, difficult work.
And so over 50 years of human endeavor from all the labs around the world,
structural biologists managed to find the structure around 100,000 to 150,000 proteins.

And they're all deposited in this database called the PDB,
and that's what we can use as a training corpus,
but also we can test our predictions.
So you can also do mutagenesis on these systems,
so you can do some genetic experiments where you change one of the sequences,
one of the residues, one of the amino acids,
and then if it's on the surface from the predictor 3D structure,
it should change the behavior of the protein.
So you can sort of check if your 3D structure prediction
saying that this residue is on the surface of the protein,
you can sort of flip that out with a genetic mutation study
and then see if that's affected the functioning of the protein.
So there are sort of various ways after the fact to sort of check whether that's right.
Okay, so AlphaFold then.
The training data you're using there is the 100 to 150,000 proteins that have been figured out.
What you have, as I understand it then, is their amino acid structures.
You have the final protein 3D structure,
and in the same way that you're setting your original system on Pong, pixel by pixel,
try to get to an outcome where you're winning points,
you are basically setting AlphaFold loose on this data
and saying try to figure out how to use the amino acid structure
to predict the protein 3D structure,
and when you do it correctly you get points.
Is that right?
Yeah, that's basically how the system works.
So you effectively have this amino acid sequence
and you're telling it to predict the 3D structure
and then you compare it against the real structure.
There's various different ways you can compare it,
but basically think about comparing where all the atoms end up being in 3D coordinate space
and you sort of measure how far it's measured in angstroms,
which is a tiny measure, right?
Basically the width of an atom.
How far away are you from the real 3D position of that atom?
And for it to be useful for biologists,
you've got to get the accuracy of that.
All the atoms in the protein, and there are many, many thousands,
within one atom width of the correct position.
That's how accurate you have to get it for it to be useful
for downstream biology purposes like drug discovery or disease understanding.
So effectively the system gets a score from the average error
it's making across all the atoms in the structure
and you're trying to get that to less than one angstrom,

less than the width of an atom on average.
So there's 100 to 150,000 of these
and then there's 100 million, 200 million proteins that we know of?
That's right, yeah.
So that's not a lot of training data actually.
You all then do something that I understand to be pretty dangerous
and usually quite frowned upon,
which is the system begins training itself on the predictions it is making.
It is generating its own data then and training itself on that data.
And there's just a new paper that came out called the Curse of Recursion
about how when AI systems begin training themselves on AI generated data,
the models often collapse.
You're basically inbreeding the AI.
So how do you do that in a way that does not inbreed your AI?
Yeah, you're absolutely right.
You have to be extremely careful when you start introducing its own,
an AI system's own predictions back into its training data.
The reason we had to do it, and this is a very interesting,
I think this is the best measure of how difficult this problem was.
So as you point out, 150,000 data points is tiny for machine learning systems.
Usually you need millions and millions of data points, right?
Like, for example, with AlphaGo, AlphaZero, we need a 10 million game,
something like that, that it played itself.
And of course, a game is far simpler than something like a protein structure in nature.
So 150,000 is very, very minimal.
I think most people assumed that there was not enough data,
nowhere near enough data.
And it turned out we had to throw the kitchen sink hat out for fold to make it work,
everything we knew.
So it's by far the most complicated system we ever worked on
and it's still the most complicated system we've worked on
and it took, you know, five years of work and many difficult wrong turns.
And one of the things we had to do was augment the real data,
which we didn't really have enough of,
use it to build a first version of AlphaFold.
And then that was just about good enough.
I think we got it to do about a million predictions of new proteins.
And when we got it to assess itself how confident it was on those predictions,
and then we sort of triaged it and cut the top sort of 30, 35%
so around 300,000 predictions and put them back in the training set
along with the real data, the 150,000 real data.
So then we had about half a million structures,
obviously including its own predicted ones,

to train the final system.
And that system then was good enough to reach this atomic accuracy threshold.
What that means is we only just about as a scientific community
had enough data to bootstrap it to do this self-distillation process.
And we think the reason that was okay was that we were very careful.
First of all, we had enough real data.
So there was a mixture of real and generated data
and that was enough to keep it on track.
And also there were lots of very good tests,
independent tests of how good these predictions were of the overall system.
Because of course, over time, new experimentalists were depositing
new structures onto the database that were past the cut-off training day
of when we trained the system.
And so we could compare how accurate the system was against those new experimental data.
So one thing you're saying here is the system is actually spitting out two things.
It's spitting out a protein structure prediction and an uncertainty score.
And that's a really actually cool thing about the AlphaFold system too,
is that in many machine learning systems, we would love the system
to not only produce the output, the prediction,
but also a measure of its uncertainty about that prediction,
a confidence measure about it.
And that's actually very important in all modern AI systems
that it would be nice if they, well, it would be very good if they all did that.
But actually very few systems do currently and it's not known how to do that in general, right?
It's an active area of research.
But we managed to do that with AlphaFold and the reason we put so much effort into that
is that ultimately I wanted this to be useful to biologists and scientists
and medical researchers, amazing experts, domain experts in their area,
but most of them would not know anything about machine learning or care, actually, frankly, right?
They're just interested in the structure of the protein so they can go and cure a disease.
And so it was really important for this particular task
that if these outputs were going to be useful to anyone, researchers down the line,
it would have to output its confidence level on the basis of each amino acid.
So it color coded it in really simple ways so that anybody,
non-expert of machine learning can understand which parts of the prediction could they trust
as an experimentalist and which other parts should they basically
probably continue to do experiments on if they wanted to know the real structure
or at least tread with caution.
And we wanted the system to really clearly output that
and those confidence levels to be really accurate.
So in effect, we needed to do that anyway for this tool to be useful downstream
to biologists and medical researchers, but we ended up using that same confidence level
to allow us to triage our own generated data

and put back the more confident, better ones into the training data.
So I want to unite this with something that people have been following,
AI are probably more familiar with, which is the hallucination problem.
So when I use chat GPT or at least different versions of it,
it is common that you can ask it a kind of question,
like tell me all about this one chemist from so and so who did such and such
and you can make it up and it can make it up,
or maybe you ask it a real question and it just makes up a citation for you.
This has been a big problem with the large language models,
still as big problem with the large language models.
And the theory is that the reason it's a problem is that they're just making predictions, right?
They don't know what they're doing in the same way that your system didn't know it was playing pong.
They just know that on the internet, on the training data they've been given,
this is the word that would be most likely to come next in a sentence.
So how come AlphaFold doesn't have this kind of hallucination problem?
Yeah, so the current chatbots today have this problem
and partly it's because if they were better able to understand the confidence
and the likelihood that what they're putting out putting is correct,
they could at some point say, I don't know, that would be better than making something up.
Or, you know, they could sort of caveat it by it might be this,
but perhaps you should double check.
And in fact, if they were able to do that, they could cross check their references.
And what they should be doing, using tools, perhaps even like search,
to sort of go, oh, actually, that paper doesn't really exist.
Just go and look it up on PubMed. It's not there, even though it's very plausible sounding.
And so in fact, the human users, I'm sure you've had experiences,
you have to go and look it up.
Just a very funny idea, right?
To make up a paper, then go search to see if the paper you've made up is actually there.
And like, oh, it's actually not.
It shows me.
Yeah, exactly.
But it would be better if it did that internally before it output that to the user.
So you never saw its hallucination.
And in a way, that's what is missing from the current systems is this,
actually, what AlphaGo does and at the systems we build,
where there's a little bit of thinking time or search or planning that's going on
before they output their prediction.
Right now, they're kind of almost like idiot savants, right?
They just output the immediate thing that just first comes to mind.
And it may or may not be plausible and it may or may not be correct.
And we need a bit of sort of, I would say, deliberation and planning and reasoning

to kind of almost sanity check what is, is that the prediction is telling you,
not just output the first thing that comes to mind.
And the kind of deep reinforcement learning systems we're known for do do that, right?
Effectively, the reinforcement learning part, the planning part,
calibrates what the model is telling it.
It's not just the first most likely move that you play in every positioning go.
It's the best move that you want.
Is this the difference between building a system that is ultimately working
with a structured data set where, at least for some amount of training data,
it knows if it has the right answer and something that is using a truly unstructured,
and I don't mean that in the technical way, but in the sort of colloquial way,
the unstructured data set of life where, you know, people just talk
and conversations don't have a right answer or wrong answer.
They're telling the entire corpus of Reddit.
Reddit is not about did you get the right answer, people just talking.
And so is what's going on here that because AlphaFold has this hundreds,
some thousand proteins in there that it knows if they're right or not.
And so it knows what it looks like to get something right,
that you can then build a system where the point is to get something right.
But when you're building these much more generalized language-oriented systems,
that that isn't the structure of language.
Language doesn't, in some internal way,
have like an input and an output where you can see that some outputs were correct
and some outputs weren't.
Yeah, I think that's the right intuition.
I mean, I think language and the way obviously on the Internet encapsulates
a huge slice of human civilization knowledge.
It's far more complex than a game and perhaps even proteins in a kind of general sense.
I think the difference is actually that in games, especially,
but also even with protein structures, you can automate the correction process.
Like if you don't win the game, then obviously the things you were planning
or the moves that you tried to make or the reasoning you did
wasn't very good to a certain extent.
So you can immediately update on that in an automated way.
The same with protein structures.
If the final structures that you're predicting have very large errors
compared to the known structures, you've obviously done something wrong.
There's no subjective decision there needed.
You can just automate that.
Of course, with language and knowledge, human knowledge, it's much more nuanced than that.
But as we talked earlier, if you hallucinate a new reference to a paper that doesn't exist,
that's pretty black and white, right?
Like you know that that's wrong.

So there are a lot of things, I think, that could be done far better than we're doing today,
and we're working really hard on increasing by orders of magnitude
the factuality and the reliability of these systems.
And I don't see any reason why that cannot be improved.
But also there are some things which are a bit more subjective,
and then you need human feedback on it,
which is why everyone's using reinforcement learning with human feedback
to sort of train these systems better to behave in ways we would like.
But of course, if you're relying on human feedback,
that itself is quite a nosy process and very time-consuming and takes a large effort.
And so it's not as quick or as automated as when you have some objective measure
that you can just optimize against.
So we talked about the exponential curves that you all have seen again and again in the gameplay systems.
And you also mentioned that the rule of thumb in proteins is it takes one PhD researcher,
their whole PhD, to figure out the structure of one protein.
So tell me the timeline then of building AlphaFold and then of beginning to find the proteins.
My understanding is that there's a kind of slow takeoff and then a real takeoff.
So what happens here?
We worked on AlphaFold, a couple of versions of AlphaFold actually.
We had to go to the drawing board at one point when we hit an asymptote
over around a four year, four and a half year period from about 2016 to 2020.
And then we entered it into this competition called CASP,
which is you can think of it as like the Olympics for protein folding.
So every two years, all of the people working on this from all the labs around the world enter this competition.
It's an amazing competition because what they do is they, over the last sort of few months,
experimentalists give them their protein structures.
They've just found literally hot off the press then, but not published yet.
So they're unknown to anyone other than the experimental lab that produced it.
And they give it to the competition organizers.
The competition organizers give it to the competing computational teams.
We have to submit within a week our predictions.
And then later on at the end of the summer, this happens all over the summer.
There's like a hundred proteins you get in the competition.
And then they reveal the true structures, you know, they get published.
And then you compare, obviously you have this double blind scoring system where nobody knew who the competing teams were
and nobody knew what the real structures were until the end of the competition.
So it's a beautifully designed competition.
And the organizers have been running it for 30 years, incredible dedication to do that.
And that was another reason we picked this problem to work on because it had this amazing competition.

So there was actually a game you could win?
There was a game we could win and a leaderboard we could optimize against.
And then when those revealed got revealed at the end of 2020,
then that day it was announced in the big conference
and the organizers sort of proclaimed that the structure prediction problem
or the protein folding problem had been solved
because we got to within atomic accuracy on these predictions of these hundred new proteins.
And so that was the moment where we knew we had a system that was going to be really useful for experimentalists
and drug design and so on.
And then the next question was how do we gift this to the world
and in a way that all these world researchers and biologists and medical researchers
could make the fastest use of to have the biggest impact downstream.
And what we realized is not only was the system really accurate alpha fold,
it was also extremely fast.
So we could fold an average length protein in a matter of seconds.
And then when we did the calculation, it was like, well, there are roughly 200 million protein sequences,
genetic sequences known to science.
We could probably fold all of them in a year.
So that's what we set out to do.
We started off with the most important organisms, obviously the human proteome.
So it's equivalent to the human genome, but in protein space.
And then we went to all the important research organisms, you know, the mouse, the fly, the zebrafish and so on.
And then some important crops like wheat and rice and so on.
Very important, obviously, to humanity.
And so then we put all of those 20 out first and then eventually over 2021, we did all of them.
And then we put that out as a database, free access database to the world,
the research community in collaboration with the European Bioinformatics Institute based in Cambridge.
Before we move on to some other work you all are doing,
one thing that as I understand it, alpha fold has spun out into now is a group under Alphabet,
the Google parent company called Isomorphic.
Tell me a bit about Isomorphic and both the sort of scientific theory there,
but also it's a very different theory of how AI could make money.
And then, you know, we're going to add a chat bot into a search engine.
So what's the business theory there?
Yeah, so alpha fold is a grand challenge in biology of understanding the structure of proteins.
The reason I thought that was so important was because I think it can hugely accelerate,
be part of accelerating drug discovery and therefore curing diseases.
But it's only one part of the puzzle.
So, you know, knowing the structure of proteins only one small bit of the whole drug discovery

process.
So there are many other really important things from identifying which proteins you should target.
So, you know, maybe through genetic analysis and genomics,
all the way to like, can we design a small molecule, a drug compound, a chemical compound
that will correctly bind to that protein and the bit of the protein you want blocked or bind to
and not anything else in your body because that's side effects, right?
Effectively, that's what makes things toxic is they don't just bind to the thing you want,
but they bind to all sorts of other proteins that you didn't want them to.
So in my view, AI is the perfect tool to accelerate the time scales of doing that.
Because that's done right now in Big Pharma in a painstakingly experimental way that takes many, many years.
You know, I think the average time of going from a target to a compound you can start testing in clinical trials
is like five, six years and many, many hundreds of millions of dollars per drug,
which is incredibly so and incredibly expensive.
And I think that could be brought down by an order of magnitude using AI and computational techniques
to do the exploration part, do that in silico using AI and computational techniques
and then only at the last step saving experimental work, of course,
very important experimental work and wet lab work for the validation step.
So instead of doing all the search, which is the expensive slow part,
you just do it for validating the compounds that your AI system has come up with.
And so isomorphic is our spin-out, sort of sister company to deep mind,
that is tasked with building more alpha folds, breakthroughs, but in adjacent spaces,
so in more going into chemistry.
So designing small molecules, predicting the different properties of those small molecules called admi properties
and making sure like we minimize things like toxicity and side effects and so on
and maximize its potential at binding to the protein that we want.
Music
So there are games, I'm very, or things that are structured a bit like games
that I'm very excited about the possibility of AI winning or getting unbelievably good at,
so drug discovery being one of them.
And then there are ones where I'm a little more nervous.
I've heard you say, for instance, that from a certain perspective, the stock market very much has the structure of a game.
And if I am a very rich hedge fund and a lot of them do algorithmic trading,
I mean, if theme park, the game was AI, then definitely what a lot of these hedge funds are doing is AI,
they've got a lot of money.
If you were thinking about a system to win the stock market, what does that look like?
I mean, there's a lot of training data out there.
Like what do you do?

Yeah, I think almost certainly some of the top hedge funds must be using,
I would have thought some of these techniques that we and others have invented to trade on the stock market.
It has some of the same properties as you say.
I mean, finance friends of mine talk about it as being the biggest game in some ways, right?
That's sometimes how it's talked about for better or for worse.
And I'm sure a lot of these techniques would work.
Now, the interesting thing is whether you just treat the stock market, say, as a series of numbers.
And that's one theory that you just treat it as a big sequence of numbers, you know, time series of numbers.
And you're just trying to predict the next numbers in the sequence.
You know, you could imagine that's analogous to predicting the next word, you know, with the chatbots.
So that's possible.
My view is it's probably a bit more complex than that because those numbers actually describe real things,
profits and losses, real companies, and then real people running those companies and having ideas and so on.
And there's also macroeconomic forces like geopolitical forces and interest rates set by governments and so on.
And so my thinking is that probably to understand the full context required to predict the next number in that sequence
would require you to understand a lot more about the world than just the stock prices.
So you'd somehow have to encapsulate all of that knowledge in a way that the machine could ingest and understand.
Well, you need to do that to fully win the game.
But to come up with local strategies that could be very profitable and are very destructive.
I mean, one, we already know that all kinds of firms have done that with high-speed algorithmic trading.
Yes.
And two, you could just imagine all kinds of, again, if you're willing to run through a very large search space
and try strategies other people don't try, I mean, you know, you could short out this company destroying this competitor
such as this other one, you could predict would go up immediately if that happened.
And you can imagine very weird strategies being deployed by a system that has the power to move money around
and a lot of data in it and is just getting reinforcement learning for making money.
Yeah, you could imagine that.
I think that as I understand it, that world, I mean, there's lots of very, very smart people working in that world with algorithms,
not necessarily the latest machine learning ones or the latest statistical algorithms.
And they're very sophisticated because obviously they're very incented to do that,

given that it's literally money at stake.
So I imagine the efficiency is pretty high already in hedge funds and high-frequency trading and other things you mentioned,
where, you know, there's already ways to slightly game the system perhaps that are already quite profitable.
And it's not clear to me that sort of more general learning system would be better than that.
It may be different from that, but it may already be easier.
There may be easier ways which hedge funds are probably already doing, I assume, to make that money.
Well, this actually brings up, I think, a pretty big question for me, which is one of the reasons I wanted to have this conversation with you about Alpha Fold
is I think people are now used to thinking about these very generalized large language models.
Inhale a tremendous amount of data, come up with these patterns, and then, you know, they're able to answer a lot of kinds of questions at a certain level,
but not a very high level rigor.
And then there's this other approach, which is building much more bespoke systems.
I mean, Alpha Fold can do something amazing in the protein space, and it cannot help me write a college essay, as best as I understand it.
And I don't want to put too much of a binary choice here, because I think I know that there is overlap,
but I do think there have been sort of two theories of how AI is going to roll out over time.
And one is we're going to have lots of different specialized systems that are tuned to do different things,
a system to do legal contracts, and a system to do proteins, and a system to check out radiology results,
and a system to help kids with their homework.
And another is that we are eventually going to pump enough data into GPT-12 or whatever it might be, such that it attains a kind of general intelligence
that it becomes a system that can do everything, that the general system eventually will emerge, and that system will be able to do all these things, and so what you should really be working on is that.
Can you talk a bit about, because I know you're interested in building a general intelligence system, can you tell me a bit about what you understand to be the path to that now?
Do we want a lot of little systems, or not little, but specialized, or is the theory here that,
no, this is right, you want it all in one system that is going to be able to span across functionally every domain?
Yeah, that's a fascinating question.
And actually, DeepMind was founded, and still our mission is to create that big general system.
That's, of course, the way the brain works, right?
We have one system, and we can do many things with our minds, including science and playing chess, and all with the same brain, right?
So that's the ultimate goal.
Now, interestingly, on the way to that goal, I always believed that we don't have to wait to get into

general intelligence or AGI
before we can get incredible use out of these systems by using the same sorts of techniques,
but maybe specializing them around a particular domain.
And AlphaFold is a great example of that, perhaps the best example so far in AI of that.
And Alpha, obviously, all our game systems were like that too.
And so what I think is going to happen in the next era of systems, and we're working on our own systems called Gemini,
is that I think there's going to be a kind of combination of the two things.
So we'll have this increasingly more powerful general system that you basically interact with through language,
that has other capabilities, general capabilities like math and coding,
and perhaps some reasoning and planning eventually in the next generations of these systems.
One of the things these systems can do is use tools.
So tool use is a big part of the research area now of these language models or chatbots.
In order to achieve what they want they need to do, they can actually call a tool and make use of a tool.
And those tools can be of different types.
They could be existing pieces of software, special case software, like a calculator or maybe like Adobe Photoshop or something like that.
So big pieces of software that they can learn how to use using reinforcement learning
and learn how to use the interface and interact with.
But they can also be other AI systems, other learned systems.
I'll give you an example.
So if you challenge one of these chatbots to a game, you want to play a game of chess or a game of Go against it,
they're actually all pretty bad at it currently, which is one of the tests I give these chatbots is,
can they play a good game and hold the board state in mind?
And they can't really at the moment.
They're not very good.
But actually maybe there's something to say, well, these general systems shouldn't learn how to play chess or Go or fold proteins.
There should be specialized AI systems that learn how to do those things, AlphaGo, AlphaZero, AlphaFold.
And actually the general system can call those specialized AIs as tools.
So, you know, I don't think it makes much sense for the language model to know how to fold proteins.
That would seem like an over-specialization in its data corpus relative to language and all the other things that general things that it needs to learn.
I think it will be more efficient for it to call this other AI system and make use of something like AlphaFold if it needed to fold proteins.
But it's interesting because at some point more of those capabilities will be forwarded back into the general system over time.
But I think at least the next era will see a general system making use of these specialized systems.

So when you think about the road from what we have now to these generally intelligent systems, do you think it's simply more training data and more compute, right?
Like more processors, more stuff we feed into the training set?
Or do you think that there are other innovations, other technologies that we're going to need to figure out first?
What's between here and there?
I'm in the camp that both needed.
I think that large multimodal models of the site we have now have a lot more improvement to go.
So I think more data, more compute and better techniques will result in a lot of gains and more interesting performance.
But I do think there are probably one or two innovations, a handful of innovations missing from the current systems that will deal with things that we talked about like factuality, robustness.
In the realm of planning and reasoning and memory that the current systems don't have.
And that's why they fall short still of a lot of interesting things we would like them to do.
So I think some new innovations are going to be needed there as well as pushing the existing techniques much further.
So it's clear to me in terms of building an AGI system or general AI system, these large multimodal models are going to be a core component.
So they'll definitely necessary, but I'm not sure they'll be sufficient in of themselves.
You've talked about in interviews I've heard you give before that you don't want to see this pursuit develop into a move fast and break things kind of race.
At the same time, you're part of Google.
They just aligned actually all of Google AI under you.
There used to be two groups, DeepMind and Google Brain.
Now it's all under your empire.
Open AI is aligned with Microsoft.
Meta is doing a lot more AI.
They're working under Yanlacun, who's like one of the founders in all this.
China obviously has a number of systems that seem to be getting better fairly quickly compared to even what we were seeing six months ago.
It does feel like a race dynamic has developed.
And I'm curious how you think about that.
I don't think it's ideal.
That's for sure.
I think it's just the way that technology is panned out.
It's become more of an engineering kind of technology or at least the phase we're in now versus scientific research and innovation,
which was perhaps done over the last decade,
and Google and DeepMind were responsible for a lot of those breakthroughs that we've discussed.
Reinforcement learning, obviously, transformers,
which Google research invented that underpin all of the modern systems.
Very critical breakthrough.
Which to give Google credit, they just released publicly.

Yes, exactly.
So they released publicly, published it available,
and everyone uses that now, including Open AI.
And so that underpins the kind of technologies and systems that we see today.
And I would prefer it if we took a scientific approach to this as a field and as a community and an industry where we were optimistic about what's coming down the line.
Obviously, I worked on AI my whole career because I think it's going to be the most beneficial technology for humanity ever.
Cure all diseases and help us with energy and also sustainability, all sorts of things.
I think that AI can be an incredibly useful tool for, but it has risks.
It's a dual use technology and like any new transformative technology,
and I think AI will be one of the most transformative in human history, it can be used for bad too.
And so we have to think all of that through.
And I would like us to have not move fast and break things like you say and actually be more thoughtful and try and have foresight rather than hindsight about these things.
We're not going to get everything right with a fast moving cutting edge technology like AI first time.
But we should try and minimize and think very carefully about the risks at each stage and try and mitigate those as far as possible while making sure we're bold and brave with the benefits.
And so we sort of have this mantra of being bold and responsible.
And I think there's a creative tension there, but it's sort of intentional between those two things.
When you look back at the technology, perhaps one of the last big technologies of the last decade or two has been social media.
I think that embodies this view of like move fast and break things.
And I feel like that has of course had huge benefits and huge growth for certain companies.
And it's been very beneficial in many ways, but it also had some unintended consequences that we only as a society started realizing many, many years later once it had reached huge scale.
I would like us to avoid that if possible with AI to the extent that that's possible.
There is also commercial realities and geopolitical issues.
And we are in this sort of race dynamic.
And what I hope is that there will be sort of cooperation actually at the international scale on the safety and technical risks as these systems become more and more powerful.
I want to talk about one benefit that you mentioned in passing there.
And then I want to talk through some of the risks more specifically.
You mentioned help us work on energy.
And we've talked a lot here about protein folding.
We've talked about the applicability to drug discovery.
I think the idea that AI could help us with clean energy is something people often hear said, but don't get a lot of details on.
But one of the systems you're building or projects you're working on is around nuclear fusion and stabilizing nuclear fusion.
So I don't want to spend a ton of time here, but just put some meat on the bones of that idea.
Can you just talk about what you're doing here and why AI might be well suited to it?
Yeah, I think AI can actually help with climate and sustainability in a number of ways, at least three

different ways I think of.
One is optimizing our existing infrastructure so we get more out of the same infrastructure.
We have a really good example of that.
Actually, we used a similar system to AlphaGo to control the cooling systems in a data centers, in these massive data centers that run all of our compute.
They use a huge amount of energy and we actually managed to save 30% of the energy the cooling systems used by more efficiently controlling all the parameters.
Secondly, we can monitor the environment better automatically, you know, deforestation and other things, forest fires, all of these types of things using AI.
So that's helpful for NGOs and governmental organizations to keep track of things.
And then finally, we can use AI to accelerate breakthrough, new breakthrough technologies and our fusion work is a good example of that.
Controlling the plasma incredibly hot, hotter than the surface of the sun.
So it can't touch the sides of the magnets and so on in these big machines called Tokamax that control this plasma, super hot plasma that is generating the electricity.
And we use AI, our sort of reinforcement learning systems to predict effectively what the shape of the plasma is going to be.
So in milliseconds, we can change the magnetic field by changing the current going in the magnets to keep hold of the plasma in place so it doesn't go out of control.
So that's a huge problem in fusion and one of the big issues with getting fusion working.
But there are also other ways I could imagine I could help in things like material design, designing better batteries, better solar panel technologies, superconductors and so on, which I think AI will be able to help with down the line.
So I want to hold that there.
And then I want to talk about a risk here, which is one of the things I see happening is ever since GPT-3 was hooked up to chat GPT.
And people could begin interfacing with it in natural language.
There's been a huge rush towards chat systems, towards chatbots.
And this is, I know, an oversimplification, but I do think there's an idea here about are we making systems that are designed to do what humans can do but a little bit better?
Are we making systems that what we have built here is something meant to seem human to humans?
Or are we making systems that are actually very inhuman, that are doing what humans cannot do because they can think in a way humans cannot think or more to the point calculate in a way humans cannot calculate?
And AlphaFold, the nuclear fusion system you're talking about, those strike me as more in that area.
And I don't want to say there's necessarily a sharp choice between the two because we've talked about the possibility of general intelligence systems too.
But there is where investment goes. There is where the corporate priorities are.
There is where, you know, the best engineers are working.
And now you have these very big companies that are basically in a battle for search and enterprise software funding, right?
They want to get subscriptions to Microsoft Office 365 up and Google doesn't want Bing to take its market share.

And one thing that I worry about a bit is that I see a lot more possible benefit for humanity from these more scientific inhuman systems.
But that the hype and the investment and the energy is going towards these more human, more kind of familiar systems that I worry are not going to be as beneficial.
And so one risk I see is simply that the business models are not going to be well hooked to public benefit.
And, you know, you said a minute ago we sort of were leaving the scientific research period of this and entering into the competitive period of this.
I think of something that kind of kept deep mind a little bit apart from it all as you're in London and you guys also always seemed a little bit more like you're on the scientific path and now you have to be on top of all Google.
How do you think about this tension?
Yeah, it's a very interesting question. I think about this all the time and you're right, there is that tension and I think all of let's say the venture capitalist world and so on has almost sort of lost their minds over chatbots, right?
And all the money is going into there. I think even in our new guys as Google deep mind, we're going to keep pushing really hard on both frontiers.
Advancing science and medicine is always going to be at the heart of what we do and our overall mission for the benefit of humanity.
But we are also going to, you know, push hard on next generation products with incredible new experiences for billions of users that help them in their everyday lives.
I'm kind of equally excited about potential of both types of things.
And so that involves us continuing to invest and work on scientific problems like AlphaFold or isomorphic labs is doing drug discovery, fusion, except for quantum chemistry, mathematics, many, many of our nature and science papers, as well as doubling down on these new types of chatbot interfaces and so on.
I don't see them as sort of human and human. It's more like the AlphaFold things are scientific tools, you know, for experts to use and enhance their work so they can accelerate their very important research work.
And on the other hand, at the moment, I think chatbots are more of a fun entertainment thing.
I mean, of course, you can do your homework on them and you can do amusing things and it's quite helpful.
But I think there's so much more to come in that space.
And I think where I see them joining together is what we discussed earlier about these general systems, perhaps that you interact with in language.
There's nothing wrong with that because language is the mode that we all can use rather than coding or mathematics.
Language is the simplest thing for everybody to use to interface with these systems, but they could call a bunch of specialized systems and specialized tools and make use of them.
And so I actually think there's quite an interesting combination to come by pushing the frontiers of both of those things.
And that's what we're planning to do going forwards.
You recently signed a letter. It was alongside Sam Altman, who leads OpenAI, and Dario Amade, who

is a top OpenAI person at LeadSanthropic.
And Lado simply says, mitigating the risk of extinction from AI should be a global priority alongside other societal scale risks such as pandemics and nuclear war.
Why do you believe there is any risk of extinction from AI at all?
Well, that letter was a kind of compromise thing. I think it's not 30 words long.
So of course, all the nuances are missing from a letter such as that.
And at some point soon, we'll put out a fuller statement about our position.
The only thing I was agreeing with there is that this technology has such potential for enormous, enormous good, but it's a dual use technology.
So, you know, if bad actors get hold of it, it could be used for bad things.
There are near term harms we have to be careful of like deep fakes and we need to address with things like watermarking and we're working on that.
And I think a lot of that will come out later this year.
And then there are technical risks, this alignment problem that we discussed earlier on how to make sure these systems do what we want.
And we set them the right objectives and the right values and they can be contained and controlled as they get more powerful.
So there's a whole series of at least three buckets of worry.
And I think they're all equally important actually, but they require different solutions.
And one of them is this longer term people think of as maybe a science fiction scenario of inherent technical risk from these systems.
Where if we don't build them in the right way in the limit when they're, you know, decades time when they're very, very powerful and they're capable of planning and all the things I discussed earlier today.
But to the nth degree, we have to be careful with those sorts of systems.
I think there is, I don't think it's likely.
I wouldn't even put a probability on it, but there's uncertainty over it.
And it's certainly non-zero.
I think the possibility that that could go wrong if we're not thoughtful and careful and use exceptional care with these technologies.
So I think that's the part I was trying to indicate by signing that was that it's important to have that debate now.
You don't want to have that debate on the eve of some kind of technology like that arriving, right?
Ten years sounds like a long time, but it's not that long.
Given the amount of research that would be required and is required and I think needs to be done to understand the systems better so that we can mitigate any risks that may come about.
Well, I think right now when people hear about extinction risk from AI,
one scenario that they've now been told to think about and more people do is the AI itself getting out of control
or turning the whole world into paperclips or whatever it might be.
But I want to talk about another here, which is more along the lines of our conversation.
So you build AlphaFold now through isomorphic.
You're building a whole suite of tools to search through the molecular space, the protein space to

better understand how to predict
the functions of and then eventually create bespoke proteins, molecules, etc.
And I think one slightly less sci-fi version of extinction or at least mass harm that you can imagine here is through synthetic biology,
through as it becomes very cheap to figure out how to create an unbelievably lethal virus and print an unbelievably lethal virus.
But in the future, it's actually not that hard for some terrorist organization to use tools like this to make something far beyond,
you think back in America, you know, however many years now, when somebody was mailing anthrax around.
If it had been very easy for that person to create super smallpox, then you get into something really, really horrifying.
And how do you think about that suite of risks?
Because you're doing more work in that space really than anyone else, I think.
And that's one of the ones that seems actually much nearer at hand to me.
Yeah, we think a lot about that and we talk with a lot of experts in government and academia about this.
And actually, before we released AlphaFold, we spent several months talking to over 30 experts in biosecurity,
biorethics, also Nobel Prize winning biologists and chemists about what we were going to release with the database and what they thought of it.
And all of them unanimously actually came back with, in that case, the benefits far outweighed the risks.
And I think we're seeing all the benefits of that today with millions of biologists around the world using it.
But look, going forward, as you get more into chemistry space, one has to think about these things.
But of course, we want to cure many terrible diseases too, right?
So we have to weigh up that enormous benefit there to society with these inherent risks that you're talking about.
And I think one of these issues is access to these technologies by bad actors,
not scientists and people, you know, medical practitioners who are trying to do good with it,
but as you say, terrorists, other things like that.
And I think that's where actually becomes a question of things like open sourcing or do you publish these results
or do you sort of how secure is your cybersecurity so you can't be hacked?
All of these questions come into play.
And I think that's where we're going to have to think a lot more carefully in the next few years
as these systems become more sophisticated about who should get access to those things,
how should that be monitored?
Can bad actors be shut down if they're using APIs very quickly before they do any harm?
Maybe we can use AI there actually to detect what are they trying to design with these systems as well?
This can also happen with chatbots too.

What are they asking the chatbots, right?
So I think there's a role for AI to play there actually as well on the monitoring side.
And so the other question though to ask is, and when I've discussed this with experts in biosecurity, there are known toxins today like you mentioned anthrax.
You can probably find the recipe for that somewhere on the internet and people could do that.
But you still need a wet lab and you still need some scientific capability.
And so those are areas which are also usually beyond a naive bad actor individual their capabilities, right?
It's not just the recipe.
How do you actually make it and then distribute it, right?
It's actually pretty difficult.
And I would argue that's already available today if you want it.
It's not that there's no bad toxins that are known.
There are some that are quite simple.
It's just not that easy to make them to the uninitiated, right?
You need a lab and you need access to it and labs can be monitored and so on.
There are still a lot of barriers.
It's not just a question of understanding the design, but we do need to think about that as well and figure out how we control that information.
Right now the systems and the kind of labs that could create the systems that could become something like general intelligence.
I mean, you could count them on two hands, right?
Across the United States, across Europe, across China.
And over time there'll be even more than that, but that's I think where we are now.
As we get closer, I mean, you were talking about how much can happen here in 10 years.
If we're getting to a point where somebody is getting near something like a general intelligence system,
is that too powerful a technology to be in private hands?
Should this be something that whichever corporate entity gets our first controls
or do we need something else to govern it?
My personal view is that this is such a big thing in this fullness of time.
I think it's sort of bigger than any one corporation or even one nation.
I think it needs sort of international cooperation.
I've often talked in the past about a kind of CERN-like effort for AGI.
And I quite like to see something like that as we get closer maybe in many years from now to an AGI system
where really careful research is done on the safety side of things, understanding what these systems can do
and maybe testing them in controlled conditions like simulations or games first,
like sandboxes, very robust sandboxes with lots of cyber security protection around them.
I think that will be a good way forward as we get closer towards human level AGI systems.
I think it's a good place to end.
So always our final question then, what are three books you would recommend to the audience?

Well, I've chosen three books that are quite meaningful to me.
So I would say, first of all, Fabric of Reality by David Deutsch.
I think that poses all the big questions in physics that I would love one day to tackle with our AI tools.
The second book I would say is Permutation City by Greg Egan.
I think it's an amazing story, actually wild story of how interesting and strange I think the world can get
in the context of AI and simulations and hyper-realistic simulations.
And then finally, I would recommend Consider Fleebers by Ian Banks, which is part of the culture series of novels.
Very formative for me and I read that while I was writing Theme Park.
And I still think it's the best depiction of a post-AGI future, an optimistic post-AGI future,
where we're traveling the stars and humanity sort of reached its full flourishing.
Thank you very much.
Thanks very much.
This episode of The Isroclancho was produced by Roger Karma.
Fact-checking by Michelle Harris.
Our senior engineer is a great Jeff Gelb.
The show's production team also includes Emma Fagau, Annie Galvin and Kristen Lin.
Our music is by Isaac Jones.
Audience strategy this week by Kristina Samilowski and Shannon Busta.
The executive producer of New York Times, Opinion Audio, is Annie Rose Strasser.
And special thanks to Sonia Herrero.