

In *What Is Intelligence?*, Blaise Agüera y Arcas explores what intelligence really is, and how AI's emergence is a natural consequence of evolution. Encompassing decades of theory, existing literature, and recent artificial life experiments, Agüera y Arcas's research argues that certain modern AI systems do indeed have a claim to intelligence, consciousness, and free will.

This talk is presented as part of a larger project on *What Is Intelligence?*, including a printed book alongside an article in the *Antikythera Journal*. It is the inaugural collaborative work of Antikythera, a think tank on the philosophy of technology, and the MIT Press.

Blaise Agüera y Arcas is a VP and Fellow at Google, where he is the CTO of Technology & Society and founder of Paradigms of Intelligence, an organization dedicated to fundamental AI research.

EDITORIAL

This lecture by Blaise Agüera y Arcas is presented as part of Antikythera's Cognitive Infrastructures Studio in July 2024. The talk delves into the nature of intelligence, drawing connections between biological systems and artificial intelligence (AI), and exploring the implications of these insights for our understanding of consciousness and the future of AI.

1. DEFINING INTELLIGENCE THROUGH PREDICTION

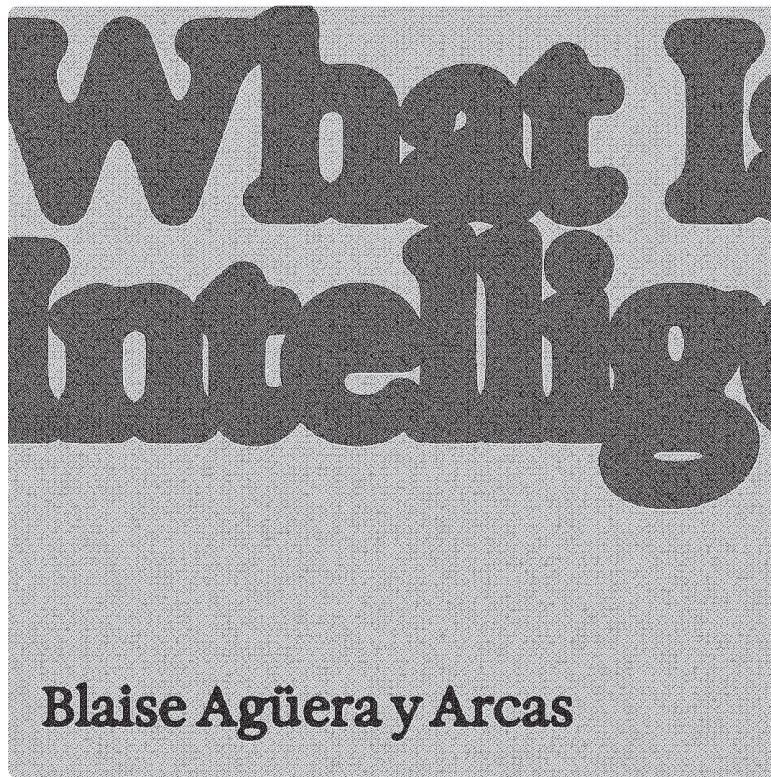
Agüera y Arcas posits that intelligence fundamentally involves the ability to predict future states based on past experiences. This concept, rooted in neuroscience, suggests that both biological brains and AI systems operate by anticipating future inputs to inform present actions. For instance, the chemotactic behavior of *E. coli* bacteria—moving toward higher concentrations of nutrients—demonstrates a basic form of predictive computation. Similarly, AI models like language predictors function by forecasting subsequent words in a sequence, embodying this predictive principle.

2. FUNCTIONALISM AND THE NATURE OF CONSCIOUSNESS

The lecture advocates for a functionalist perspective, asserting that intelligence and consciousness are defined by their functions rather than their physical substrates. This view aligns with Alan Turing's ideas, suggesting that if a system performs the functions associated with intelligence, it should be considered intelligent, regardless of its composition. Agüera y Arcas extends this argument to consciousness, proposing that systems capable of modeling themselves and others—exhibiting theory of mind—may possess a form of consciousness.

3. EVOLUTIONARY PERSPECTIVES AND MAJOR TRANSITIONS

Agüera y Arcas contextualizes the emergence of AI within the framework of major evolutionary transitions, such as the development of multicellularity or eusociality. He suggests that AI represents a new evolutionary phase, characterized by complex interdependencies and collective intelligence. This perspective emphasizes the role of mutual prediction and cooperation in driving the evolution of intelligence, both biological and artificial.



Thank you all so much for being here. And, thank you for the almost decade long collaboration. Thank you to all of the technical staff who have, changed my t-shirt three times. To find one that the tape would stick to, and that didn't-- Anyway. Let's get to it.

So, 'What Is Intelligence?' is both the title of the talk and the title of a book that, hopefully, Antikythera and MIT Press will be publishing in about a year, and then will start to get serialized online in fall. The original title was an homage to Erwin Schrödinger's little book 'What Is Life?' The original idea was to write a book that was just as short as Schrödinger's about what intelligence is. But it's going to be a little longer than Schrödinger's book, I'm afraid. And this talk might be a little on the long side, too, but I'm going to try and race and so we're going to go on a ride together.

00:55

What is Intelligence?

This is not what the cover will look like, but I'm hoping that this font with its kind of, you know, erosion and contraction thing that James Goggin, designer that I've been working with for a number of years and I really geeked out about. So hopefully the font will make it— and the font as part of the presentation as well.

I'd like to say a few words about my own story, because it's probably relevant. I grew up in Mexico City. I was always a little bit of an alienated kid, and spent a lot of time, as I imagine some of you did as well, playing with computers. And so, you know, this was my main friend. Other than my cat. And these machines were simple enough that anybody could understand them. You know, even an eight year old kid. And, that's not the case with any computer nowadays. Even the simplest computer that we use in real life is beyond the understanding of any single person. So, I feel, in a way, like my generation is a bit of a special one that has seen that transition. As Benjamin Labatut has put it: from a world that we understood to a world that we can no longer understand. Not individually anyway.

So this is how I learned how to program on my Texas Instruments 99/4A and then on Commodore 64 and then IBMs and various other machines. And, I then studied physics at university and ended up, after university, going to the Marine Biological Laboratory in Woods Hole, Massachusetts. And getting into computational neuroscience there, working with Bill Bialek.

And MBL is a kind of storied institution. It really has played an integral part in developing the basic theories of computational neuroscience, of how it is that neuroscience and function and computing weave together. A bunch of key discoveries have been made there, often using marine animals that live just off the shore in that part of the world. But I sort of went away from science for a while, and founded a little startup which got acquired by Microsoft, and then ended up working on a lot of proto AI sorts of technologies in computer vision. And, this stuff was based on a lot of handwritten code that, in this case, reverse engineered the 3D features in photos and reconstructed 3D from them.

And then, over time, what started to become clear is that neural nets were going to take over from these handwritten, or good old-fashioned AI approaches. And when it became clear that that was happening, in the early 2010s, I left Microsoft and joined Google because that was, at the time, the epicenter of that sort of work.

Benjamin mentioned in his introduction DeepDream, which was the work of Alex Mordvintsev on my team at Google. It was one of the first visual generative AI models. And it certainly wasn't generating anything photorealistic. It looked more like a really bad acid trip, but it was a first. And the main work of my team at Google since 2014 has been building a lot of on-device AI models of various kinds, things like the Now Playing feature on Google Pixel that recognizes songs and many other AI features.

But, the one that I think is most relevant for where things ended up going was Gboard, the Google Keyboard. So it's a simple piece of AI. It's just something that predicts what the next word is likely to be, based on what you're typing on the keyboard, and therefore, speeds up typing a little bit. It's a sequence model, so it uses the statistics of letters and words to predict which letter or word is likely to go next based on which ones have come before.

We all knew that it was going to suck. On the one hand, you know, keyboards are not very fast to type on. At least for people like me who haven't learned to do it with their thumbs properly on a phone. So, you know, it only needs to be a little bit okay to do better than one finger typing. But, we didn't think that you'd be able to make a statistical model out of next word prediction and get anything that would actually, really be able to predict next words because the problem has unbounded hardness. And I want to explain a little bit about what I mean.

So, the statistics of language, of letters and of words, is in theory very simple. It's just a giant table. If you look at the first order table, it's just the letter frequencies. So these are the frequencies of the letters in English. Many of you probably know the first few: E, T, A, O, I, N and so on. That's easy. That's probability of letter, not conditioned on the past.

Now, this is the two-dimensional version of that. This is probability of letter one, comma, letter two. That is, you know, a sequence of letters. And so it's a little more complex. It's a bigger table. And you can actually generate words with it. You can generate fake words with it by just starting with a random letter and then picking, probabilistically, what the next letter is going to be. So, you know, these are some deepfake words generated using only this bigram, or two letter kind of model. They're not great words, but they're words like 'felogy'. So, you can think of this as P of next letter given previous letter. And if you extend that—if you generalize it to words—then you can say, well, you know, P of next word given previous word. It's exactly the same kind of table, just a lot bigger. Now it has maybe 30,000 rows and columns, one for every word.

So, you know, if the first word is humpty, we all know what the next word is. I don't know if I'm allowed to do audience participation in this talk, but. 'Dumpty'. Okay. Helter? 'Skelter.' Yes. Yin and? 'Yang.' Right.

Now here we've moved into P of next word given previous words. So, this is an even bigger table. This is now three dimensional in word space. Or we can have lots of previous words, like after Ballmer's retirement, the company elevated \_\_\_\_\_. All right. And now we start to realize that in order to fill in the next word, it's not good enough to just think about histograms of words that have been used in general. We actually need specific knowledge about the world—that after Ballmer, it was Satya Nadella who became—that it's Microsoft that we're talking about. And that when Ballmer retired was Satya who became the CEO. Or in stacked pennies, the height of Mount Kilimanjaro is \_\_\_\_\_. Well, now, in order to predict the next word, you need to be able to do math and convert units, and figure out how thick a penny is. Or, when the cat knocked over my water glass, the keyboard got \_\_\_\_\_. Now you need common sense reasoning to understand that it's wet that the keyboard got and not dry or broken or something else. Or a shipping container can hold 436 twelve-packs or 240 24-packs, so it's better to use \_\_\_\_\_. And now we'd have to understand, well, you know, it's likely that you want to put more things in a shipping container as well as do the math. So, in other words, these things get arbitrarily hard as the context gets bigger. After the dog died Jen hadn't gone outside for days, so her friends decided to \_\_\_\_\_. Now I don't know what words come next, but in order to get them right, or to get them plausible, you need to have theory of mind about Jen, what she might have been feeling at this point.

So, in other words, this is why we thought that the Gboard had a roof or a limit on how good it could get. Because, histograms are good enough for predicting next the word when it's Humpty Dumpty, but they're not good enough for predicting these kinds of words or these kinds of words. You really need understanding.

So, the real shock was that when we just took the same kinds of models that we had been building these smart keyboards with and made them really big and trained them with very large corpuses, they actually did get all of those kinds of completions right. And this was a shock because, you know, we knew that those kinds of completions were 'AI complete', as the term of art goes, that they require understanding of every possible kind, and therefore we thought: there's no way that this simple trick would actually work, because some magical extra fairy dust would surely be needed in order to make them intelligent. But making them bigger, apparently, was that fairy dust. And that was a shock.

But maybe it shouldn't have been as much of a shock as it was, because, you know,

back when I was still doing computational neuroscience, there was there was an idea going around—and it had been going around for a long time—that the brain really is all about computing this very same function.  $P$  of the future given the past;  $P$  of what is next, given the previous stuff.

The modern neuroscientist and researcher that this idea is most closely associated with is Karl Friston. This is his paper, 'A theory of cortical responses', from the early 2000s. But basically his claim has always been that the brain is just a next token predictor. That's why we've got brains. And, in some sense, this theory actually goes all the way back, at least to Hermann Helmholtz in the 19th century, if not farther back.

The question is, if you have something that predicts the future given the past in a string of text, and does it pretty well, is that counterfeit intelligence? Is it just pretending to be smart? Or what if it's the real thing? And, the argument that I'm going to be making here is that it's the real thing. This is what brains do. And when we build models that can do this well, that is what intelligence is. And I'm aware that it's a very controversial argument, but I guess what I would say is: if you know math, you can pretend not to know math on the test and get a bad score. But if you don't know math, you know how many students have wished that they could pretend to know math and get a good score on the test? Like it just doesn't work the other way around. And I think the same is true for intelligence in general.

11:48

Counterfeit Intelligence

This idea is one that Turing was also pretty enthusiastic about. And that's where the Turing Test came from. So, the Turing Test was first introduced—although obviously, he didn't name it after himself, he called it The Imitation Game—in 1950, in this very famous paper in *Mind*: 'Computing Machinery and Intelligence.' 'I propose to consider the question, can machines think?' And, the basic idea of the Turing Test is really simple, which is: if it walks like a duck and quacks like a duck, it's a duck. And, so it was based on the idea that if you can interrogate the computer and ask it various questions in order to probe its intelligence, and its answers convince you that it's intelligent, then there is no other meaning to what it is to be intelligent. It's not like there's some fairy dust that is or isn't on the inside.

And this is an idea that sometimes has been called functionalism. Functionalism, meaning that intelligence is defined by function in the mathematical sense. What goes in; what comes out. It's about what that function does, not about what implements it or what it's made out of. Intelligence is as intelligence does. And a function in mathematical terms is just a computation.

So, it's no coincidence that Turing thought this about intelligence. Turing was also, in many ways, the father of computer science and defined the whole terms of how we think about computation, what computation is.

So, some function  $f$  of  $x$  or some function probability of future given past. And I'm going to take things a little bit further than Turing did, and propose that this functional definition of intelligence applies to life itself as well: that life is also a certain kind of function, and moreover, that it's the same kind of function that intelligence is.

So, why would life bother to compute the function  $P$  of future given past? This is the first thing that I guess I would like to convince you of, that life cares about this function. Well, here is a time lapse picture of an *E. coli* cell. And you can see how quickly they reproduce. That's minutes on the bottom. So, you know, one cell turns into two, turns into four. We have exponential growth. This reproduction is obviously critical to what life is. If life didn't reproduce, then it would no longer exist in the future, right? So if life had been there at some point in the past and it didn't make more copies of itself, then it would be a roll of the dice or a flip of the coin, whether it still lives at the next moment in time; and once it's dead, then it won't exist in the future anymore. So, the fact that life exists now, because it existed in the past, has to mean that it is bringing more of itself into existence at a given moment. That's why reproduction is so fundamental to life.

Now, in order for *E. coli* to make more *E. coli*, they've got to eat. And one of the ways that they figure out what to eat is by chemotaxis. They're attracted to sugar and other nutrients. This is a sugar crystal in the middle of this microscope. And you can see that when you release *E. coli* into the flow cell they swarm around the sugar crystal because that's the material that they need in order to make more little *E. coli*. How does it work?

So this problem of how chemotaxis works is one that Howard Berg, the great biophysicist, studied at length in the 1970s and wrote a bunch of really beautiful papers about, but the basic discoveries are that *E. coli* have these filaments, called flagella, that come out of their cell membrane. And there are motors. It was the first rotary bearing that was discovered in nature. And the motor can either rotate clockwise or counterclockwise. So when all of the flagella are rotating in one direction, they kind of twine together and they form this bundle that lets it swim forward. It swims forward more or less. I mean, it's diffusing around, so after a certain amount of time, it's going in a random direction, but it can swim forward, or if those motors reverse direction (or a couple of them reverse direction, then these flagella will fly apart and the bacterium will tumble in place. And when it tumbles, it randomizes its orientation.

So, kind of like those old radio-controlled cars. Again, this is probably a reference that will not make any sense to anybody who is younger than I am. But, you know, they could only go forward, or go back and turn. It's kind of like that, except that the turn is random. So it's got a one bit output: forward or randomize direction.

So, how can they swim toward sugar with only this control that says either swim forward or randomize direction? Well, there is a basic computation that Berg showed that you could do, which is, roughly speaking, if the concentration of sugar is increasing, keep swimming. And if the concentration is decreasing, tumble, randomize your direction. If you do that, then statistically, you will end up swimming toward the goodies.

So, there is a function being computed inside the bacterium. A bacterium has a very, very complex inside. And in particular, the mechanism for doing all this chemotaxis stuff has been described by Jeff Stock, a biophysicist at Princeton, as a 'hair brain'. It's a kind of brain made up of proteins of various kinds. And this brain processes the inputs of chemicals docking and undocking with the cell membrane. And its output is this one bit: run or tumble. So it's a function. The input is molecular docking events, the output is flagella spin clockwise or counterclockwise. The membrane is an interface.

So how it computes—what the platform is, what kind of machinery does the computing—really doesn't matter. And in fact, all of life generally has various redundancies and different pathways that it can follow for metabolism and for computation, such that really all that's being selected for is that the behavior is right. If one pathway doesn't work, if one kind of food isn't present, then you just use a different pathway. You compute in a different way. And this is what I mean by functionalism, that what matters is the relationship between what it senses and what it does. That's what's being selected for by evolution. So, you can think of it as probability of action given stimulus. Evolution is an algorithm for learning that function: probability of action, given the stimulus. I will show you.

So here is some code to implement a really simple kind of toy bacterium. And the toy bacterium has some source of food, quote unquote, which is the red spot. And, here, the bacteria aren't moving. So you can see that when we nucleate this kind of computational plate, they all die off because they're only getting food when the red spot is on top of them, and they're not succeeding and following the red spot, so they don't get enough food to reproduce.

Well, now, instead, we'll say they have this table of action versus stimulus. What you see on the right-hand side is individual tables of action given stimulus for each of those bacteria. And the rule is if it gets enough food, it can split and copy its table. If it doesn't get enough food, it dies. That's it. And when you do that and you run it for a while, then evolution happens. Because basically the only bacteria that live, and that reproduce and that regenerate their table, are the ones whose table successfully implemented an algorithm that would let them follow the food around. So they learned chemotaxis.

Evolution is a learning algorithm. What it's learning is the function, and the function is the life form. The organism is this function, P of action given stimulus. If you run it a bunch of times you get many different kinds of behaviors. So, you know, these are all different solutions, if you like, to this problem of chemotaxis. Some of them are very conservative. They follow the food really closely. Others are very aggressive. They do a lot of exploration rather than exploitation, and they die frequently. It doesn't matter. They all win because, in the end, these different personalities or solutions are not really trying to maximize any given score. What they're trying to do is just to keep playing. This is an infinite game, and in an infinite game, the only prize is that you get to keep playing.

So, this is an active inference problem, just like the one that I described. You know, what Friston said brains do. Because even though I've written it P of action given stimulus, the thing is that actions that you take determine the future stimuli that you'll encounter. And also, those actions are themselves future stimuli. One of the very first things that an organism can recognize as an input is the things that it does itself. When it decides to rotate the flagellum, this way or that way, that's also a signal that it can use; it can know what it did as well as whatever it encounters on the outside. Everything that happens on the inside of the membrane is a signal, just as much as what happens on the outside.

Like hunger, for instance. Hunger is a signal or a concentration of something on the inside of the cell, just like concentration of food is a signal on the outside. That's the origins, by the way, of dopamine and serotonin.

So a viable model works, in that it successfully predicts itself into existence in the future. That's what life is. Life is a probability of the future given the past that is viable: that successfully predicts its own existence in the future.

And that's the only way for that function to be dynamically stable. If it decides to predict itself into nonexistence, well, then it won't exist. Or if it's a crappy model and it doesn't successfully predict its own future existence, then it also won't exist. So what I'm saying is something almost tautological. The only things that are around are the things that successfully predicted their own existence in the future, in the

past. I hope that makes sense. I want to switch gears a little bit now and talk not just about why life computes this, but how and what we mean by computation at a little bit of a deeper level.

So this P of future given past is a function that has to be computed. And this is a physical model of a Turing Machine, which is Turing's abstraction for how all computations work. It's a pretty fun project by Mike Davey from 2010. It's actually like a physical Turing Machine, actually made. I don't know how many of you are familiar with what a Turing Machine is, but basically, it's an infinite tape that a head can move back and forth on, one step at a time, and either read, write, or erase characters. So this is that, rendered literal, with a little dry erase marker and an eraser—and a Turing Machine running the Turing Machine! But never mind.

The original term 'computer' referred, of course, to people. This is the first mention of 'computer' in a newspaper. It's in the New York Times in 1892, and it was an advertisement for somebody to come and actually do this kind of work in the Almanac office and so these were the original computers, of course. And when Turing thought about this tape moving left and right and following a table of instructions, it was these women that he was thinking of.

The Church-Turing thesis, which is this idea of universality of computation, held that any way of doing a computation is equivalent. If you just define a very, very simple machine and you can make an isomorphism or a mapping between that and what a person could do, then the person and the machine can compute the same thing. Also, that there are many different ways of making a computer, and that it doesn't matter which one you pick. You can always translate between any one and any other one.

That doesn't mean that computation is logical or rational or optimal. It doesn't even mean that it's entirely predictable. A classic Turing Machine doesn't have any randomness in it, but Turing imagined models of computation that did have randomness in them. And in fact, those kind of random models were super important in the early history of computing.

This was a very early computer, the MANIAC, in 1952. It was the one that was used to make the first hydrogen bomb calculations. It was done using the Monte Carlo Markov Chain method. And the Monte Carlo is named after the Monte Carlo Casino. It's there because there's a random number generator at the heart of this algorithm, as there is at the heart of many algorithms—even quicksort, probably the most famous algorithm of all time. Just for sorting numbers, it uses random numbers in order to decide which pivots, or which numbers to check against which other ones.

So randomness was understood to be a really important part of doing algorithms from the very beginning. This is Turing on the right at the Ferranti Mark I computer, one of the very earliest desktop sized computers. Turing insisted that the Ferranti Mark I should have an instruction (it's called /w) that would generate a true random number, using a noisy resistor—you measure the physical noise in the resistor, and generate a number. It didn't make very good random numbers, but it was a random number generator.

I'm dwelling on this because in the popular imagination, when we say that something is a computer or something computes, we tend to imagine HAL 9000 or Data from Star Trek. There's this idea that that means hyper-rational, that that means that everything follows exact rules, that there's something, lawyer-like, or precise, you know, lacking ambiguity, lacking anything like emotion. This is a holdover from an old idea in good old fashioned AI that has nothing whatsoever to do with computing.

Although it does trace all the way back to Leibniz—in many ways, the even earlier father of computer science than Turing. Leibniz made many fundamental contributions to the theory of computing in his day. Among other things, there are many inventors of binary, but insofar as anybody could be said to have invented binary, I'd say Leibniz is a good candidate for that. He designed this medallion, in which he introduces binary, with this epigram, 'God made everything (one) from nothing (zero)'. He believed there was something divine about binary, and that if we could express problems in the right language or 'characteristica universalis', then it would allow for computation to be able to resolve any question. You know, is same sex marriage good or not? Or, who should be the Pope? By literally just computing. That was what he imagined would be possible.

He said "if controversies were to arise, there would be no more need of disputation between two philosophers than between two accountants, for it would suffice to take their pencils in their hands, sit down with their slates, and say to each other, let us calculate [calculemus]." He was imagining a Data, or a HAL 9000, that would be able to literally just calculate its way in some absolute sense to the answer to any problem. Now, we know this is bullshit. Nowadays we know this is not possible to do. But it was a very persistent idea.

George Boole, the inventor of Boolean Algebra, also had it. And he also believed that the way brains worked fundamentally was Boolean, was based on logic. And so perhaps if we're not so rational, that means that there's some errors in our brains or errors in our calculations.

His widow, Mary Everest Boole, in 1901 wrote, “nearly all the logicians and mathematicians ignored the statement that the book was meant to throw light on the nature of the human mind and treated the formula entirely as a wonderful new method of reducing to logical order masses of evidence about external fact.” In other words, Boole’s book about logic was supposed to be a book about neuroscience just as much as it was about math.

Ada Lovelace had similar ideas. She, as I’m sure you all know, was the collaborator of Charles Babbage, working on the Difference Engine and the Analytical Engine—which had it been built, would have been the world’s first computer, in the 19th century. Sort of steampunk magical thing that was, I think the best piece of scholarship on the Analytical Engine is Sydney Padua’s comic book, *The Thrilling Adventures of Lovelace and Babbage*. It was published back in 2015.

Close to the end of her life, which was cut far too short by cancer, Ada wrote “I have my hopes of one day getting cerebral phenomena into mathematical equations; in short a law or laws for the mutual actions of the molecules of brain. The grand difficulty is in the practical experiments. In order to get the exact phenomena I require, I must be a most skillful practical manipulator in experimental tests, and that on materials difficult to deal with: the brain, blood, and nerves of animals.” 1844. Had she lived to be older, I don’t doubt that she would have done this. This was all happening around the same time as *Frankenstein* was getting written by the way.

So 99 years later, that calculus of nervous activity that she was imagining was actually created by Warren McCulloch and Walter Pitts, in 1943. This paper ended up being enormously influential both in the history of computers and in the history of neuroscience. But it also marked the last point at which those two fields were really joined at the hip. They diverged afterward.

McCulloch and Pitts imagined that neurons were logic gates. When they looked at neural nets like these... these are three neurons with excitatory and inhibitory connections with each other. That loop is an inhibitory connection. And the shapes of the neurons are the shapes of pyramidal cells in the cortex. They were imagining that they literally just implemented logical or Boolean operations. So the brain was a neural net that implemented, literally, a logical calculus.

When Ramón y Cajal drew his diagrams of cortex like this, that’s the surface of the cortex on the top, you can see that the neurons look kind of like those pyramids. Those are pyramidal cells. When McCulloch and Pitts drew their neurons, they drew them the same way. If you’re an electronics person, you will recognize that that circle for the inhibitory synapse looks exactly like the NOT symbol that goes on the end of logic gates when they’re drawn in circuit diagrams. So our drawings of logic gates, I’m pretty sure, are literally just drawings of neurons. I’ve had a hard time finding that connection in the literature, in the 1940s, but I’m pretty sure that’s what it is.

Now, there was another direction that things went as well. It was also neural in character. This was Frank Rosenblatt in 1958, working on the Perceptron. The Perceptron was a machine that started off with random connections and random weights. The weights were actually implemented with little volume knobs and motors on the volume knobs that had a feedback loop. The goal was to make something that would be able to recognize different shapes, different letters or triangles from circles or what have you, and it would feed back errors onto the motors connected to the volume knobs and change the weights in this network.

When the Cornell Aeronautical Laboratory, where he was working, publicized this work, they talked about it as the design of an intelligent automaton. “Introducing the Perceptron, a machine which senses, recognizes, remembers, and responds like the human mind.” Rosenblatt’s diagrams of the perceptron and of the brain, as you can see, were identical. He was literally trying to make a brain.

Well, it didn’t work all that well at the time because we didn’t have very big computers. But Hubert Dreyfus, who was critiquing the logical paths that AI had taken in 1972, what we now call good old fashioned AI, wrote “even if the brain did function like a digital computer at some level, it would not necessarily provide encouragement for those working in CS or AI, for the brain might be wired like a very large array of randomly connected neurons, such as the perceptrons proposed by the group Minsky [this is Marvin Minsky] dismisses as the early cyberneticists. Such a neural net can be simulated using a program, but such a program is in no sense a heuristic program.”

In other words, the kind of AI work that was based on logic, Data-like stuff or HAL 9000-like stuff. By 1972, it was already clear that that was a program that wasn’t going to work. It couldn’t even do things like recognize objects. The Perceptron could, but it represented an opposing camp that didn’t imagine that what we were doing in our brains was logic. Still a function, but a function with randomness and a function that didn’t involve computing logical propositions the way Data and Spock and so on do.

So, there was a fork in the road. AI and computer science went down one road, cybernetics went down the other road. And, well, went off into the weeds for a few decades. Cybernetics sort of faded from the popular imagination for many, many years. The AI that was imagined on the left is what we now call good old fashioned AI. It never worked out, although everything about computers really comes from that

path nowadays. Spreadsheets, and missile trajectory calculations, and your iPhones and Androids, and so on. That's all a product of the left hand path. Modern AI is a product of the right hand path. There's a straight line from cybernetics and what Frank Rosenblatt was doing to what AI is doing nowadays. But it was in obscurity for a long time.

However, remember that they're both functions, and they can both be implemented on any kind of computer. That's Turing's equivalence principle. As Turing wrote in 1950, "the fact that Babbage's Analytical Engine was to be entirely mechanical will help us to rid ourselves of a superstition. Importance is often attached to the fact that modern digital computers are electrical, and that the nervous system is also electrical. Since Babbage's machine was not electrical, we can see that this use of electricity cannot be of theoretical importance. If we wish to find similarities, we should look for mathematical analogies of function," not of how it's done. This is nowadays called 'platform independence' or 'multiple realizability'. Anything that an Analytical Engine could compute, a modern computer could compute, doesn't matter whether it's steampunk or electrons or something else.

So if intelligence and life is about the evolution of functions that predict their own future, how do those functions arise in the first place? How does computation itself arise? I'm going to now show you another series of experiments that I think really shed light on that.

This is a very recently published work. We put the paper up on the 27th of June just a few days ago. The title is Computational Life: How Well-Formed, Self-Replicating Programs Emerge From Simple Interaction. Here's how it works. We were interested in figuring out how self-replication could emerge out of nothing. And we used a Turing-complete language called Brainfuck to figure it out.

Why did we use Brainfuck? Well, because this language—and apologies, this is actually the French Wikipedia page translated into English, because it happened to have a minimal Brainfuck program—this is 'Hello World' in Brainfuck. Brainfuck only has eight instructions. It's a very, very small language and very, very hard to program in. But it's Turing complete. You can write anything in it. You can write Windows 95, you can write the Doom video game ... but nobody will be able to reverse engineer it. And in fact, probably nobody will be able to write it either. It's a very, very difficult language, to either read or write, but in theory it can do anything. So—that program that you see will print out 'Hello world'. I'm sure you can all see that.

This is what the set of instructions actually do. For those of you who are kind of on the computer science side and are interested: one instruction increments this data pointer by one; one decrements by one; one increments the value at the data pointer; one decrements the value of the data pointer; one outputs a byte; one inputs a byte; and finally, there are these, open and closed bracket, which are for loops. The open bracket will jump to the close bracket if the value at the data pointer is non-zero, and the closed bracket will jump back to the open bracket if the value was zero. Sorry, I probably got this the wrong way round. That's it. That's all of the instructions. That's everything in Brainfuck. And you can do everything with that.

Okay, so what did we do? Well, we began with a soup of 8192 random programs of length 64—64-byte long Brainfuck programs. Actually, to call them programs is really overstating the case, because there are only eight valid instructions. And these are really random ASCII strings, which means that only 1 in 32 of those bytes are even a valid instruction. So, that means that on average, every one of those strings will only contain two valid instructions. Everything that is not a valid instruction just gets skipped over when it runs. So what we do is we pluck two of those 64-character long programs out of the soup at random, stick them end to end, and run. Now, this code can modify itself, its input and output is the string itself. But of course, its likelihood of doing anything useful or interesting is virtually zero. When we pluck these two things at random out and we run, the likelihood is that nothing will happen at all. But after it runs, we pull them back apart and drop them back in the soup. And that's it. That's the whole story. There's also a little bit of mutation. There are cosmic rays that come now and then and flip one of these bytes to some random other value.

This is a very unpromising situation, right? In the beginning, when you stick two of these guys together and you run, the average number of instructions that executes is unsurprisingly two. That's what that number is up there—two ops per interaction. All right. Are you all ready to see what happens? I'm going to run it on my machine. Actually I recorded it with my phone on my computer. But you'll see what happens in real time now.

After a while—because computers are really fast nowadays, it's not that long—something begins to happen. Something pops up. These are programs that are well formed. Out of nothing but random interaction between strings, after a few million interactions, we get well-formed Brainfuck programs. It actually takes quite a lot of cleverness to go and figure out what the hell they're doing.

They're reproducing. That's what they're doing. There are 5000 copies of that top one, 297 copies of the next one, and so on, and the average number of ops per interaction after these 8 million or so interactions is 4784. In other words, they're doing serious computational work. They're well-formed programs, and they're doing real stuff. And, if you look at the difference between time equals zero and time equals 5

million, you can see that something has really happened here. So this is pretty cool. It's like self-writing software.

Here's a really interesting way of looking at what happens. There are 10 million dots on this plot. Every dot represents one interaction. The x axis is time and the y axis is how many operations—how much computation happened in that interaction. The average at the beginning is two. But, as you can see, there's a certain special moment. Here, it happens right at about 6 million interactions, when suddenly the state of the soup changes. It's like a phase change—like freezing, or some kind of phase transition. If you take the complexity of the soup and you calculate that—here I'm using a method called Kolmogorov complexity, it's easy to do. You just take the whole soup, all of the strings in it, and you ZIP them. You make a ZIP file out of them, and then you take the ratio of the ZIP file to the original. That's the Kolmogorov complexity.

Why does this matter? Well, because if it's random, then when you try to ZIP it—I don't know if any of you have ever tried to ZIP a random file, but basically nothing happens. It stays the same size. You can't compress random noise, but when there starts to be structure, then it becomes compressible, and you see that right at that phase transition, that's exactly what happens. The complexity suddenly drops dramatically. Not to zero, not to something very low, but to something much lower than it was. That is statistical regularity. It represents a new phase of matter, if you like. The phase before 6 million is like a gas and the phase after 6 million is what you might call machine phase, or computronium, or life.

Now, that transition is inevitable. It happens at a random time with any given run, but it always happens.

I'll show you a much more beautiful visualization of this, made just a few days ago by Alex Mordvintsev, the same guy who did DeepDream. He's, among other things, a very brilliant web coder. He did something way cool, which is to make a visualization of these programs that would interact with each other in a grid. They interact with their nearest neighbors, and you can see what the little program was. He actually did this not in Brainfuck, but in Z80, which is the same instruction set that my old childhood computer used. You can see life nucleating and forming, and species spreading and, you know, sometimes wiping each other out, sometimes mutating and recombining and forming new kinds of life. Each of these pixels is a program, and the programs go from noise to reproducers. This is all, going to be online in about two weeks. It actually runs on the GPU, right on the web page. You can kind of try it out.

I think that's what life is, and if you think about it, understanding the underlying language or the underlying computational mechanism is irrelevant to the life itself, because—remember—everything is platform independent. What matters is the function. And if you change the language, there's always a way of translating what happens into any other arbitrary language. So, there is some sense in which life is independent of physics, because all that does is give you the box of tools that you've got in order to implement functions. And if the box of tools were different, you could still implement the same functions or any other ones. So—that's life. So much for life.

Let me try and make clearer why computation is an attractor. In other words, why things go from a state of not computing to a state of computing. Actually, John von Neumann pretty much nailed this one back in the 1950s. He invented the field of automata theory—another of the fathers of computing. He wrote a beautiful paper, 'The Theory of Self-Reproducing Automata', which posed the following question. Suppose that you've got some thing, some machine that paddles around on a pond full of parts, such that the parts are sufficient for it to make a copy of itself. All it's got to do is harvest them and put them together. How could such a machine exist?

Of course, what he's really asking is the same question Schrödinger is asking, which is what is life, right? Because what I've just described is reproduction. What he realized is that what you needed was a slight generalization of the Turing Machine, and, in particular, a generalization that made it really physical. There has to be a tape. There has to be some list of instructions for how to put stuff together in order to make yourself—the instruction manual for making yourself. And there has to be a machine A that will copy that tape. And there has to be... no, I've got that the wrong way round. There's got to be a machine B that can copy that tape, and there's got to be a machine A that can walk through the instructions and assemble using those instructions. As long as the instructions for making machine A and machine B are both on the tape as well, then you have a life form.

Now, what is so remarkable to me about this paper is that he wrote it before the structure and function of DNA had been understood. But of course, DNA is exactly that tape. DNA is a Turing tape for doing exactly what von Neumann was imagining. His machine A is a ribosome, and his machine B is DNA polymerase, which copies the DNA tape. That's exactly what we've got. We've got a tape on the inside of all of our cells that says how to build the cell, including the proteins that themselves build anything that is on the tape. And that copy the tape. That's all you need. And you can put any extra stuff on that tape that you want to build other kinds of functions too. It looks like pulling yourself up by your bootstraps, but it's not. All you need is this machine A and machine B.

What I find so interesting about this idea of von Neumann's is that it requires computation. You can't reproduce without a computer, without—literally—a Turing comput-

er. There's got to be a loop that walks through the tape and builds and stops when it's done. There have to be conditional instructions that say: either build this or build that. Either add this molecule or add that molecule. In other words, reproduction is not possible without computation. And that's why computation is an attractor. Because if you want to have something that builds copies of itself, it must compute.

You can't reproduce without computation. And reproduction is the most basic requirement for predicting yourself or yourselves into continued future existence in an environment where inaction would allow entropy to erase you. That's why reproduction is active inference, too. It's the same, in other words, as intelligence, as we were talking about earlier.

What that implies is that pretty much any universe that has a source of randomness and can support computation will evolve life, and presumably intelligence, because they're the same thing. This is a conjecture. We're working on the theory behind this, but I think that the conditions are very, very minimal. And it kind of has led me to believe that there is probably life all over the place because it is just this process of matter seeking its most stable state. There's an old joke about DNA being the most stable molecule in the world because, you know, granite will wear down, but DNA makes copies of itself, so it'll exist for a long time.

All right. Now, why does complexification happen? Once you've got bacteria, they're reproducing. How do we go from bacteria, to eukaryotes, to us, to AI? Why do things get more and more complicated over time? Well, it's because every replicator creates niches for more replicators, including sub replicators. So, in other words, once you've got a machine A and a machine B and a tape, and those things are reproducing everything on the tape, well, now that tape becomes an environment or an ecology in which other replicators can take root. The tape might be mostly random. In the case of Brainfuck, for instance, there's a little machine A and machine B, but there's a bunch of noise or random stuff that is also getting copied whenever those things get copied. Well, what's to say that you can't get another little replicator, inserted in that random noise that will maybe replicate itself—even multiple times per run. And that's exactly what happens.

So this is a not very comprehensible graph showing you how multiple replicators arise, in this Brainfuck kind of process. And that's why when you look at the number of bytes that actually code for instructions as opposed to for noise, over time, you see that this transition happens way at the left side of this plot, right? When it goes from close to zero to starting to rise—it continues to rise over time. And that continuation—the fact it becomes more and more computational over time—is because of sub replicators starting to emerge. In other words, it's symbiosis.

Wherever life can take root, life is more dynamically stable than non-life. And so, any place that can will develop life. And the more life there is, the more places there are that life can take root. Because every bit of life is an ecology for other kinds of life.

That's how you get things like mitochondria, making their way into archaea in order to form eukaryotes. Mitochondria are their own little replicators operating inside the replicators that we know as cells. They are just like those multiple replicators in BFF. It's how you get a lot of ribs in a snake, right? Those ribs are replicators, too. And the snake is an ecology, where one rib could beget more ribs. It's like a little Adam and Eve story, but, you know, in the snake instead of in the people. That's also why there are many cortical columns in humans. That's also a replication process inside us.

Eörs Szathmáry and John Maynard Smith wrote a beautiful paper in *Nature* in 1995, a review article about their theory of major evolutionary transitions. These are the moments when everything changed on Earth—and there aren't a ton of them. I don't think their list is exactly right. They don't think their list was exactly right, either. They revised it a couple of times since, but it's short. Things like replicating molecules to populations of molecules and compartments, unlinked replicators to chromosomes, prokaryotes to eukaryotes, protists to animals. Their last one is primate societies to human societies via language.

It is a short list. And these are the really big changes. Evolution is gradual, and then these moments of big transition are really moments of symbiosis. There are the moments when things that are replicating on their own suddenly find it advantageous to replicate jointly, even if it becomes impossible for them to replicate on their own. This thing that they make together is a more robust replicator than what they could be on their own.

If you go back in the fossil record to the Ediacaran, at that time, most things that we had on Earth were very simple life forms, they looked kind of like this. We had some things like jellyfish. These were motile cells cooperating via mutual prediction to make animals.

What do I mean by cells cooperating to make animals? Consider how fireflies synchronize their flashes when they swarm. I'm sure you've all seen fireflies. There are some species of fireflies that synchronize. Why do they synchronize? This will take us too far afield. But it might have to do with concealing their flash from predators, while at the same time attracting mates. In any case, it's beneficial for these fireflies to synchronize. How do they do it? Well, by prediction. They're making a model of the fireflies around them. They're predicting when they're going to flash, and their

action is also to flash at the same time. They can't do that without eyes. Their eyes are what allow them to see what the firefly is doing on the other side of the clearing so they can synchronize their flash with that firefly.

If you're a jellyfish, or if you're a bunch of cells that are kind of like amoebas, or muscle cells on their own that are going to join together in order to start to pulse like this jellyfish, how do they synchronize with each other? They don't have eyes. They can't see what's happening, tens of centimeters away on the other side of the jellyfish. They have to grow tendrils that reach over to the other muscle cell in order to feel what it's doing, in order to synchronize their own action with that other action. And that's what they did. This is how distributed nerve nets evolved. They evolved as sensory mechanisms, allowing a muscle in one spot to sense what a muscle in another spot is doing, in order to coordinate their actions.

When we go a little further in the evolutionary tree to the bilaterians, they start to swim more or less forward, and are symmetric across an axis. Now, there is a special part of the animal that is going to experience the world first, before the other parts. That part is the front. If the muscle cells are growing sensory organs, little tendrils to feel what it's like somewhere, they're going to especially want to feel what it's like at the leading end, because their synchronization activity is going to want to be informed by the changes that that leading edge is going to experience. That's how brains evolve. Brains evolve essentially as knots of sensory neurons for the muscles elsewhere in the body of the animal.

One of the things when the reason that I'm really belaboring this is because it's actually the very opposite of the way many people think about what a brain is. We tend to think of a brain as being a homunculus, as being the boss. You know, the brain controls the body. What I'm saying is, no, the body has sent its sensory organs into the head, and that's what the brain is. It's just the sensory endpoints of the muscles. So, the brain is not a homunculus at all. It just so happens that that's the most interesting place for all of the muscles to sense what is going on.

Now, in the Cambrian, after the Ediacaran, animals become much more complex and we get these really bizarre looking things, like like this guy, Marrella, full of spiny bits. You can see exactly what's going on, which is that they're starting to eat each other. So, these are animals that are getting very aggressive and learning how to do predation. Increasing the cooperation of the cells within their body is the way they can increase the competition between animals. Cooperation within allows for competition without.

So, there's a predation arms race. The faster they can coordinate their own movements, and the better they can work together in order to attack or defend, the greater their stability is going to be in evolutionary terms. And that's not only powered by mutual prediction of cells inside the body, but also powered by the mutual prediction of those animals with each other. If I'm the predator, I will want to predict which way my prey is going to go in order to, you know, jig left when it jigs left, and eat it. And if I'm the prey, I'm going to want to predict my predator so that I don't jig left when it thinks I'm going to jig left. There's a kind of arms race there because the better I'm able to predict the other, the better I will do. I also want to predict the other's model of me, and I want to predict what it thinks I'm thinking it thinks as well. So higher and higher orders of prediction, and higher and higher orders of theory of mind, are what the game is all about. Whether you're trying to cooperate or compete, it doesn't matter.

This is exactly the same thing that happened in World War Two. We began to cooperate. The Axis and the Allies began to cooperate at massive scale in order to destroy each other. It was a kind of Cambrian explosion of warfare. This period was also, not coincidentally, when cybernetics was born. Cybernetics was all about how to control antiaircraft weapons, to predict where the plane is going, in order to fire where it's going to be in the future. P of future given past.

The father of cybernetics, Norbert Wiener, took this on as his motivating problem. He was very much inspired by these turret guns, which were thought of at the time, even back then, as mechanical brains, working in metal boxes: computing devices aim guns and bombs with inhuman accuracy.

This is a Palomilla or 'moth', the robot that Wiener and his students built in 1949 at MIT. That's Palomilla going down a hallway. It's doing a very simple cybernetic calculation. It's just following the light. Or if you reverse the motors, it turns into a bedbug and runs away from the light.

When Wiener and his collaborators wrote the paper Behavior, Purpose, and Teleology in 1943, they formulated this whole predictive hypothesis pretty cleanly. They talked about behavior as being active or non-active; active behavior as being non-purposeful or purposeful; purposeful behavior as being non-feedback driven or feedback driven, which is to say purposive; teleological and purposive behavior as being non predictive or predictive, meaning extrapolative, which also tells us to think about the future; and if it's predictive then it could be at higher and higher orders as I've described. So, Wiener also got the predictive brain hypothesis pretty much right in the 1940s, just like Karl Friston would, decades later.

58:10 The Birth of Cybernetics

It's a straight shot from here to the Perceptron to unsupervised large language models and modern AI, as we have today. All we've been doing, as we make our models bigger and bigger, is increasing the order of the prediction. The model becomes bigger, the order of the calculation becomes higher, the ability to predict becomes better and better. We've written a paper recently about theory of mind in large models showing that, as the models get bigger, their theory of mind improves. That's not a coincidence.

Now, during human evolution, we have seen the sizes of our brains explode—over the past seven or so million years, and especially in the last 3 million years. Why? Well, it's exactly the same reason. Because we are trying to out-predict each other. There's lots of work showing that if you are able to better predict others in your social environment, you will have more mating opportunities, you'll have more political success.

Nicholas Humphrey, the computational neuroscientist, first formulated this theory, that predicting each other was at the heart of intelligence explosions like the one we underwent. In 1976, he wrote "Of all the animals in the forest"—this was when he was when he was visiting Dian Fossey's gorilla lab in Africa—"Of all the animals in the forest, the gorillas seem to lead much the simplest existence. Food abundant and easy to harvest, provided they knew where to find it. Few, if any, predators, provided they knew how to avoid them. Little to do, in fact, and little done, but eat, sleep, and play. And the same is arguably true for natural man." He was thinking about the wealthy hunter gatherer hypotheses that were in vogue at the time. But that's true.

You know, it's not like you have to be smart in order to find food, or in order to just fight with or hide from a predator. But you really have to be smart in order to outthink your own conspecifics. When you begin to either compete with your own conspecifics or, at a group level, when others of your own species are living in different bands, and the band that can cooperate at the largest scale survives better, right? Either that group level cooperation or that individual level advantage it's what causes those intelligence explosions.

You can really see that when you look at work from Robin Dunbar in the 90s, at the relationship between brain size and sizes of bands. This is where the Dunbar number comes from. This is a bunch of different kinds of monkeys and apes, and humans. You can see the relationship between the size of the neocortex and the average social group size that they're able to support. A bigger brain means a bigger troop; a bigger troop means a more intelligent troop. That's what scaling cooperation (or scaling competition) is all about. And you scale cooperation through theory of mind.

Now, I don't know how many of you have seen this movie of Richard Linklater's, *Before Sunrise*, from the 90s? It's very romantic. It involves two really hot young people who meet on a train, coming into Vienna. And, they kind of have, you know, maybe the hots for each other. They get off together, they spend the whole night wandering the city together. They agree that this is just like a one-off. They're not going to—this is pre-cell phones—they're not going to exchange contact information. They're going to just go their separate ways at the end of the night.

But there's this very moving scene at the end. Well, moving if you're into indie films like this. They suddenly realize, as she needs to get on the train and go, that they're really into each other and they want to see each other again. They have a very short time to figure out how they're going to see each other again in six months. And they decide, okay, six months, here, this platform, at 6:00, on such and such date, we're going to meet here.

Why is this interesting? It's interesting because if you think about the bags of molecules that are Ethan Hawke and Julie Delpy, they are highly unpredictable, right? They're in a complex universe. They're super non-linear in physics terms. There is no way that you can predict anything about where either of them will be in six months' time. As physical systems, they are about to diffuse, about to blur, about to undergo a kind of Heisenberg uncertainty principle. But psychology lets the two of them predict where they're going to be six months from now. (They actually don't get together in six months, but never mind.)

What I'm saying is that psychology offers a more powerful predictive model than quantum physics. Which is a very backward way of thinking about things relative to the usual framing in which you think about physics being the rock bottom and psychology just like a flaky pseudoscience on top. No: theory of mind is actually much more powerful.

It's especially powerful when you consider that the whole point of living systems is to, in some sense, be competitive with each other with respect to their predictability. When you look at fluttering moths, they're trying their damndest to be unpredictable so they don't get eaten by the cat. So, the flight of a moth is chaotic. It introduces randomness, in order to blur even faster than it otherwise would. Both that unpredictability of life, that critical instability of living systems, and the fact that we are predicting each other—in a way, those two phenomena work together to create what we call free will.

Free will is a combination of critical instability, of the fact that we're always on the edge of chaos, always could go one way or the other—life seeks that razor's edge.

It seeks that razor's edge so much that a few words whispered into your ear, right? A little tiny bit of vibration in your eardrums can change where your entire body will be in six months, right? This is a critical instability. It's a combination of keeping ourselves on those edges where we could jig right or jig left, whether it's to avoid the cat, or to make a decision about our own futures. A combination of that and high order theory of mind, including a theory of our own mind.

A self is a theory of your own mind. When you think: "how will I feel next week if I make this decision or that decision?" you are modeling yourself, and you're selecting from among many possible futures based on a model of yourself.

When you amplify randomness of the same kind that Turing introduced into the Ferranti I computer, in order to be able to explore all those various different counterfactual universes, what you're doing is essentially allowing the future to determine the past or allowing a model of the future to determine what actually happens. It's this backward causality that gives us the willies when we think about the idea of psychology following from physics. But this is why it works. It works because we are actually modeling ourselves in much the same way that a Turing tape in von Neumann's replicator models the physical structure of an animal. We have mental structures that model our mental selves in the future.

So, free will doesn't violate physics, even though it sure looks like it does. And this is what Schrodinger, I think, meant in *What Is Life?* when he wrote "living matter, while not eluding the laws of physics as established up to date, is likely to involve other laws of physics hitherto unknown, which, however, once they have been revealed, will form just as integral a part of the science as the former."

So, what is consciousness? Well, it's a self-modeled you that selects from one of those futures. When you say "I do a thing" or "I'm aware of a thing," what you mean is that you are modeling yourself, being aware of a thing. You're modeling yourself in the same way that you model somebody else when you're deep in conversation with them. That's what the self is. It's not an epiphenomenon or an illusion. It's necessary in order to be able to pick among counterfactuals and make these long range decisions like, am I going to be on this train platform in six months time?

But how singular, how unified is it? We all have this sense—this is where the homunculus feeling comes from—that we are like this one unified thing. And we're not.

I'm sure many of you have heard of these split brain experiments in which, back in the old days, there used to be a medical intervention in which people's brains were literally cut in half when they had very severe epilepsy, in order to prevent electrical storms from propagating from one hemisphere to the other one. When that happens, people generally recover quite well. But there are some very startling results, one of which is that if you show different things to the left and right visual field, then the right hand can only respond to what the left hemifield sees, and the left hand can only respond to what the right hemifield sees. Speech, generally, is controlled by the left part of the brain. So you'll only report on things that you see on the right hand side. It's pretty weird because, as the experimenter, you, in a way, have to be modeling two different people in there.

But one of the things that I've always found so interesting about these split brain experiments is that you can never get somebody with a split brain to admit it. You know, they never come out and say, "I feel like there's another person trapped in here." It just doesn't happen. Even though, at times, one hand will be buttoning the buttons and the other one will be unbuttoning the buttons. And you get phenomena like: you show some text saying "get up and walk" to the side of the brain that doesn't control language. And the person will get up and start walking. The experimenter asks, what are you doing? And they say, "oh, well, you know, I'm headed to the kitchen, I was thirsty," right? So you make some bullshit up, and everybody's very weirded out.

But the thing is, what each hemisphere is doing is auto-completing what the other has started. They're computing the same thing. P of future given past. They have a unity of purpose in the same way that a rowing crew has unity of purpose when they're rowing a boat together.

This is another wonderful example. This is somebody with blindsight. This is a case in which the entire visual cortex has been destroyed. But it turns out that we have another visual pathway through the deeper parts of the brain, that dates back to the time of the frogs and lizards. And often people with blindsight can actually do feats, like visual navigation of an environment—walking around and avoiding the obstacles—even though they report that they can't see. You say, well, just pretend that you can see. Just do it anyway. Just walk—and they'll do it.

How is that possible? Nicholas Humphrey claimed that it's because, somehow, the visual cortex is conscious while these subcortical visual regions are not conscious. But I don't believe that's the case at all. It just so happens that the subcortical regions that are able to see are not connected to the part of your brain that does the talking.

We are, it turns out, all in the same situation as these blindsight patients and split brain patients. Petter Johansson and his collaborators have done a series of experi-

ments, called choice blindness experiments, in which you can basically fool people using sleight of hand into justifying decisions that they didn't make. The classic ones are, for instance: you're shown two faces, and you're asked which one is the more attractive one. You make a decision. They're swapped a little while later, and you're asked to justify why you made the choice you did not make. People's fluency and ability to answer is unchanged based on whether it's the one they chose or the one they didn't. Only about, I don't know, 30% of people even notice that the change has happened. And they just spout bullshit immediately. We're all stochastic parrots! It's the same thing if you ask which flavor of jam you like, or a series of political questions.

People just make stuff up. That's what we do. We have the illusion of being singular, because we really are like a fractal of mutual predictors that are all kind of on the same team. You're all on team because wherever you're going to be in six months, you know, all the different parts of your brain are going to be in the same spot. That's all it is.

This is called the interpreter. The part of your brain that does the bullshitting or the lawyering. But really, it's just that: all those parts of your brain are in the same boat, just like a rowing team.

There's no homunculus, there's no center. It's just a fractal cascade of mutually predictive models.

1:12:21 Intelligence is...

Intelligence is predictive. Intelligence is social. Intelligence is fractal, meaning it's social within as well as without. Intelligence is diverse. The fact that these different predictors are, if you like, feeling different parts of the elephant, seeing the left side of the visual field versus the right, that's really important. If everybody had the same inputs, then they wouldn't be adding anything to the whole. It's the fact that they disagree—that they're not all aligned, to put it in the terms of Benjamin's talk from last year—that makes the whole greater than the parts. And of course, they're symbiotic. This works out when that diversity of points of view and that internal turmoil makes for a unified whole that is better able to predict itself into existence in the future than the individual parts would be able to.

All right. I know I've been going for a long time. We're in the homestretch. This brings us to Transformers. The transformer is the model architecture that has been really taking over the world in the last several years. Some of you are probably wondering, okay, is there anything special about transformers? Are they real intelligences or are they not? Well, they are just predictors. That's what unsupervised pre-training is all about. Is the transformer it? Is it the final model? Well, it is being used not only for chatbots nowadays, but for audio and for video and for robotics and so on.

But it has some very obvious deficits relative to the kinds of predictors we've got in our own heads. They don't have state, meaning they just have this context window. They don't have any durable internal state beyond that. They don't have long-term memory, and they don't learn as they operate. And those are all big shortcomings. We don't get trained and then run afterward. We are always running and always training, except maybe when we're sleeping, and that's very different for these kind of models.

The lack of state can lead to some very funny stuff. Like, for instance, there are cases where you can set a math problem for a transformer-based chatbot, and it might get the right answer, but for the wrong reasons— gets the right answer and you say, okay, how did you do that? And it'll come up with something that wouldn't give the right answer. Then, a lot of people like Gary Marcus are like, well, it's bullshit. It's obviously not really smart. Well, this is just the interpreter at work, right? It could be the case that that cascade of neural activations did the right thing for the right reason. But then when it goes to explain, it doesn't have all those activations anymore, right? When it's emitting the tokens to make that explanation, it's like a person, saying which face they liked or which kind of jam.

Now our brains are recurrent, we have introspection. We have thoughts and plans and inner monologue. We have a stream of consciousness. Transformers apparently don't have that, although I want to point out that, in just the past year or so, models like QuietStar or Drafts in Gemini or just chain of thought prompting are pretty important steps toward having something like a stream of consciousness or inner monologue.

So QuietStar, for instance, is really just a switch that the transformer has got that says, am I going to say this in my outside voice or in my inside voice? Very, very simple idea. Or Gemini Drafts allow for multiple drafts to be written, and you're not shown all of those drafts unless you ask. You just see the final draft. That's what our stream of consciousness is. It's when we don't just blurt everything out that is going through our heads. (Although I sometimes do.) And it's precisely by not disclosing all of that inside voice stuff that you can present a unified front to the outside based on potentially disagreeing or working things through, over time.

Chain of thought prompting, I'm sure many of you know about. Without chain of thought, you can pose problems like Roger has five tennis balls, he buys two more cans of tennis balls. Each can has three. How many tennis balls does he have now? The answer is 11. And then another question is posed. All of this is the prompt. The

prompt is there in order to have the model predict what the next answer is going to be. And it blurts out, the answer is 27. It's wrong. Don't bother working it out. It's wrong.

Now, with chain of thought, all you do is to change the answer that is given as an example to one that actually works it out step by step. Roger started with five balls. Two cans of three tennis balls each make six tennis balls. Five plus six equals 11. The answer is 11. Just like your math teacher tried to teach you to do, right? Don't just blurt out the answer. Work it out step by step. And then, when it has to work it out, it does work it out, step by step, and it gets it right. No different from us. So, step by step reasoning is basically the interpreter—in particular, when you are not saying that stuff on the outside. Our stream of consciousness just happens to include to-kens that we don't say aloud.

There's nothing uniquely human in this. You know, we often think we're the only ones that have an inner life. We have inner monologue, other animals don't. I think that is so far from the reality. Take, for instance, the jumping spider. So, this little guy—it has a very, very serialized kind of brain because its brain is so small, it doesn't have room for the kind of parallelism that we have in our brains. So each of its little eyes is like a telescope that can only see a few pixels, but it will plan out an attack on another spider, sometimes over the course of an hour. It may go around and break line of sight for 45 minutes and sneak up, and then finally attack. And it plans all of that out using a chain of thought over long periods of time, including over long periods of time when it doesn't have the sensory environment that it needs in order to make those inferences. Even seeing what is going on requires that it scan around, almost like a scanning microscope, to reconstruct using state.

What I'm trying to say is that thinking slowly is easier with fewer neurons. It's actually only when brains got really big that we could even do the parallel processing that allows us to make rapid, unconsidered actions. We have things backwards when we think that reasoning or working things through over time is the really advanced stuff. No, the really advanced stuff is being able to add to the step by step reasoning lots and lots of parallelism, maybe a lot of cortical columns doing it at once, or maybe a single step that allows a bunch of processing to happen at one time.

That chain of thought, that ability to serialize a kind of state, is also what libraries give us. And this is one of the key reasons that as a society, we have a superintelligence greater than our individual parts. It's only by having a chain of thought at societal scale that is recorded in a durable way—nowadays, of course, with electronics as well that we are able to act in ways that are not just about the state that we're able to retain in real time individually.

These are all areas of ongoing research: state, long term memory, online learning. I think I'm going to skip in-context learning. But suffice it to say that the boundary between learning and inference is not nearly as firm as one would think. Basically, when you have a bunch of stuff in the context window a transformer is actually learning from it, as it turns out. But, what is not happening yet is that that learning isn't yet getting baked back into the model. And this is a very important area of ongoing research.

1:19:44 Ongoing Research

All right. So, let me just pose the big, million dollar question. Is there anything it is like to be a chatbot? Here of course I'm paraphrasing Nagel with his 'is there anything it's like' or 'what is it like to be a bat' question. This is the big C question. The consciousness question.

Well, you are a model that models itself. That's, I think, what it means for you to be a self, for you to be conscious. Any being that models other models, including itself and at nth order, you know, me modeling your model of me, and so on, will appear to be conscious. And that's why when you have a conversation with an advanced chatbot with a large model, you will certainly have the appearance of consciousness in that interaction: because it has theory of mind. We've run all of the standard tests—Sally-Anne tests and so on—for theory of mind, and they have a pretty good theory of mind, because that's part of what you need in order to do a good job of predicting the future from the past, when that past includes lots of other beings (like us) writing texts to each other and stuff.

So, if we take Turing seriously with functionalism when he asks, Can machines think?" then we have to acknowledge that it's the function that matters. Any social being that models other models, including itself, and at nth order, will appear to be conscious. And I don't think that that 'appear to' necessarily belongs in that sentence. I think if we differentiate between 'appears to' in every way and 'is' we are leaving science behind, and that's really the fundamental lesson that Turing was trying to teach us with the Turing test.

So ... periods in history. "Wait, but why" guy did a very funny sort of essay and comic—I think a rather profound one—a number of years ago, in which he talked about what was going to happen with AI intelligence explosions. Haha. That's adorable. The funny robot can do monkey tricks. Wait a little bit. And we got there. I think that's kind of where we are now.

There are a lot of people, in the AI community and elsewhere, talking about AGI, artificial general intelligence, as some kind of a threshold out in the future. I don't

understand why they are saying that. The term “artificial general intelligence” was invented years ago in order to distinguish real intelligence of the kind of Turing would have recognized from narrow intelligence—that is to say, like solving CAPTCHAs or recognizing speech from text or something. That’s what narrow intelligence is. General intelligence is intelligence that can do anything. And that’s what we’ve got. I think if you Google search “artificial general intelligence” for results only prior to 2020, it will be clear that what we have today is artificial general intelligence. We’re just kind of like the dog that caught the car and doesn’t know what to do now. So we keep moving the goalposts for what general intelligence is forward and forward, to the point where it becomes something mystical or undefined. This is where we are.

And this is a case that Peter Norvig and I made in October of last year, in Noëma. It’s a major transition in evolution, just like the other ones on John Maynard Smith and Eörs Szathmáry’s list. And, you know, I would say that there probably have been multiple ones that actually postdate theirs. There’s human eusociality, which is not just humans, but the technosphere. There’s electronics, which I think were a pretty big transition around 1920. There’s the internet, which was probably a pretty big transition. And there’s AI now, and I think this one is the biggest one in a very long time.

These major transitions in evolution, they can certainly be disruptive, but they are fundamentally creative acts. It’s not like the bacteria stopped existing when eukaryotic cells came on the scene. It’s not like individual humans stopped existing when we became eusocial, when we began cooperating at much larger scales. And I don’t see any reason to believe that humans are going to stop existing when AI comes. I actually think we’re just having a bit of a freak out, because we’ve had the illusion for a long time that we are top dog. And even the sense of us being top dog is predicated on a wrong supposition, which is that we are individual rather than this big collective superorganism.

In what sense is a person the smartest thing on Earth? You ask the average Manhattanite how the toilet works and they don’t know, right? Or how the toaster works or anything. Right? So individually, we’re very weak as intelligences. It’s only because we cooperate together to form this collective superorganism that you can even talk about a human intelligence in any modern sense.

And I’m not even sure that it’s human. I think that it includes the wheat, it includes the cars and trucks, it includes the cats and dogs and the cows, and all of the other components that come together to make this eusocial lichen or superorganism that makes us up today.

Now, does that mean that we have nothing to worry about? No. Climate collapse could be really bad. We might be dancing on the brink of a truly catastrophic event. We don’t know, because our models of the planet are not good enough for us to be able to tell what kind of interventions or screw-ups of the ecosphere might turn us into Venus. We just don’t know.

Also, with much more certainty, we know that we’re very close to nuclear apocalypse. We have enough nuclear weapons now on Earth to, if not sterilize the planet, certainly wipe ourselves out. And they are armed. And that is insanity. The fact that, that we are having conversations about AI existential risk when we literally still have these weapons pointed at each other maddens me. I cannot understand how an intelligent species or an intelligent superorganism can have its finger on this particular trigger and be thinking about anything else as an existential risk.

So, I think less about X-risk than about “Y-risk,” or something I don’t have a good name for yet. But there are certainly risks that we face now. We face risks in our economic system, in our democracy, and our systems of governance, none of which are well-suited to an AI symbiosis of the sort that is emerging right now. We face energy and infrastructure challenges, and we face some challenges in purpose and identity, which are what Benjamin was getting at with his five stages of grief. These are important. They’re not species enders, but they’re disruptive, for sure. And we need to figure them out. I think that there is something pretty great on the other side of that. In order to survive at planetary scale and thrive at planetary scale, we need a planetary scale intelligence.

I’m not saying we don’t have work to do, but I think that on the one hand, we need to do things like disarm all of the nukes and get rid of the actual existential risks. And, on the other hand, develop the intelligence to be able to operate at planetary scale effectively.

And I’m going to end there. Thank you all so much. I know this has been a really long talk.