**Scott Aaronson:** 0:00

The real issue with a lot of these quantum annealing approaches is that, even theoretically, even supposing that you had a perfect quantum computer that was as big as you wanted it to be, we don't actually know that these approaches would be classical, annealing the basic way that a quantum state can change over time. Well, there are two ways: One is measurement and the other one is unitary transformation. For me, the single most important number to look at is actually the two-qubit gate fidelity, which means like with what degree of accuracy can I apply a single two-qubit entangling gate of my choice?

**Craig Smith:** 0:37

Hi, I'm Craig Smith and this is Eye on AI. In today's episode, we dive deep into the intricate world of quantum computing. We've got Scott Aronson, a computer science professor and director of the Quantum Information Center at the University of Texas, Austin, to help demystify quantum computing. Scott has worked in the field for almost 25 years and is here to disentangle the confusion, debunk the hype and enlighten us on where quantum really stands today, from quantum's role in AI to its practical implications in the real world. Prepare for a deep dive into the quantum realm. Let's get started. This episode is sponsored by Crusoe Cloud. High performance cloud computing or low environmental impact? Crusoe Cloud was built because the innovations of the future need both. Crusoe Cloud is a scalable, clean, high performance cloud, optimized for AI and HPC workloads and powered by wasted, stranded or clean energy. Crusoe offers virtualized compute and storage solutions for a range of applications, including generative AI, computational biology and rendering. Visit crusoecloud.com that's C-R-U-S-O-E C-L-O-U-D dot C-O-M to see what climate-aligned computing can do for your business.

**Scott Aaronson:** 2:18

So I'm Scott Aronson. I'm a computer science professor at University of Texas, Austin, and I direct the Quantum Information Center here. I'm actually on leave for a couple of years right now. I'm working now with OpenAI on theoretical foundations of AI safety. So that's actually not about quantum computing, but I still run my research group at UT. You know, half a dozen PhD students. So I've been working in quantum computing I guess for almost 25 years and my interests, a lot of them, have been about the limitations of quantum computers, what we can't do even with a computer that we don't have, and you know. But I've brought into a bunch of other topics quantum advice, quantum proofs, quantum software, copy protection, quantum money, you know. And then computational learning theory applied to quantum states. You know, learning properties of quantum states using few measurements. And then you know the connections between quantum computing and the black hole information problem, which has become a major topic over the last decade. So, yeah, those are some of my interests and yeah, happy to, you know, I mean, I also write a blog. I've been writing it since 2005. And you know this wasn't the original intention. It sort of became a main clearinghouse for, you know, countering quantum computing hype, you know, just out of necessity, because no one else was doing that right. And you know like whenever there would be some breathless announcement, you know I would start getting emails and calls from journalists asking me to respond to it, just because you know no one else was writing a blog like that one, and so then I would try to do that. And so I've learned something about it. You know how to talk about this to a broad audience. It's still like I get calls from journalists asking okay, I need you to explain what quantum computing is in one sentence. I've been trying for 20 years, you know. I think I can do it in 10 minutes.

**Craig Smith:** 5:00

Yeah, well, they've now.

**Scott Aaronson:** 5:02

But not in a sentence.

**Craig Smith:** 5:04

They've now got a GPT chat. You can do that, yeah, of course I've tried asking for it.

**Scott Aaronson:** 5:10

It gives you know, you know, right now I would say, a mediocre explanation. And what especially hurts my feelings is that even if you ask GPT to, you know, explain quantum computing in the style of Scott Aronson, you know, it still doesn't do that great. Now that I'm at OpenAI, maybe I can get someone to.

**Craig Smith:** 5:32

That's right. I'm the model, yeah exactly. Yeah, this Scott Aronson version says so. Yeah, and I'm interested about the safety work I heard you at on the collective intelligence, I guess webinar talking about that work, so we can get to that. But to start, and you know, actually ironically, I'm in the house I live in my father-in-law's house. He's now back in Japan and he won a Nobel Prize in quantum tunneling.

**Scott Aaronson:** 6:16

Leo.

**Craig Smith:** 6:17

Asaki. He won with Gustavson and was at 73. They split the price through anyway, and he worked at IBM Research, which is a couple of towns over. So I'm embarrassed that I don't understand.

**Scott Aaronson:** 6:37

Are you close to Yorktown heights?

**Craig Smith:** 6:39

Yeah, exactly, we're in a chapel.

**Scott Aaronson:** 6:41

Yeah, I know that, I know that area. Yeah, yeah.

**Craig Smith:** 6:45

And, as a matter of fact, I want to go over and I've been invited to come over and look at their quantum computer.

**Scott Aaronson:** 6:50

Yeah well, I mean, you've got you know a bunch of experts in the subject who are right there. I mean Sergey Bravier. Actually, my former student Patrick Rohl is now working there, so I'm sure they'd be happy to talk to you. Yeah, yeah.

**Craig Smith:** 7:06

Okay, but so I guess the first question that puzzles everybody is you've got all these quantum computing companies, both, who are offering quantum services, you've got clouds that offer quantum computing and you have companies that are commercializing quantum computers. So what exactly do those computers do? Can they do it?

**Scott Aaronson:** 7:39

Well, the main use of them right now is just to do experiments, to understand what these particular kinds of quantum computers do. Right, I mean so to a first approximation. And here I realize that I am up against a tsunami of hype, you know, asserting the opposite, okay, but you know I am. You know, because you're asking me, you know I'm going to tell you that what I think is the truth that probably none of these devices that are available right now are going to give you a speed up over a classical computer for any practical problem. Okay, yeah, that might change in a few years. Okay, I think that that's a true statement as of right now.

**Craig Smith:** 8:25

So, beyond the research aspect, I was talking to a guy at a company call who's talking about optimizing financial models and he just lost me, so what?

**Scott Aaronson:** 8:48

I'm wincing here because there are possibly hundreds of startups right now that are about that, about using quantum computing for optimization, especially machine learning, and finance and things like that. I think that this is at least as things stand right now. This is almost all phony. So eventually, if you have a full, scalable quantum computer with error correction, we do know of algorithms that would give you modest advantages for these types of problems. In particular there is a famous quantum algorithm called Grover's algorithm. That was discovered in 1996. And it basically lets you solve just about any search problem in roughly the square root of the number of steps that a classical computer would need. So getting a quadratic speed up, that could be a big deal, but that's not an exponential speed up, like if I had 10 to the thousandth power possibilities to search through. Well, the square root of that is 10 to the 500, still a quite big number. But that could eventually push forward the frontier of what sizes of problems we could handle. But I think that it will be a long time before any Grover-type speed ups are actually a net win in practice, after you have to factor in all of the expense and all of the slowdown that comes from just needing to run an error corrected quantum computer in the first place. And so then, what does that leave? Ok, well, so there are these quantum algorithms that get exponential speed ups over anything we know how to do classically, but most of those are for two specific categories of problems. One of them is breaking cryptographic codes, and this was the famous application that put quantum computing on the map in the first place 30 years ago. A Shor's factoring algorithm Factors large numbers exponentially faster than any known classical algorithm, and it, in its variants, could break most of the public key encryption that currently protects the internet. So that would be a big deal. It's not clear that that's a positive for humanity. It would mean that we all have to upgrade to new methods of encryption and then, if that goes well, then we'd kind of be right back where we started. And then the second application. I think this is the one where most of the real promise of getting a practical benefit from quantum computers is, and certainly a benefit anytime soon is just in quantum simulation. Okay, this was the original application that Richard Feynman had in mind when he gave a now famous speech 42 years ago proposing the idea of a quantum computer that would have this programmable device for simulating any chemical reaction, any material. If you're trying to design better batteries, better solar cells, you can now simulate the dynamics of all the interacting electrons. Even though the wave function is this enormous object and it has a number of parameters that grows exponentially with the number of particles, a quantum computer would be sort of the purpose built device for handling that, because it itself would have the same exponentiality. So those are the two big applications where we're quite confident that you could get an exponential speedup code breaking and quantum simulation. But then that's not the reason why there's been this rush of venture capital and money for startup companies and so forth over the last decade. That's mostly been because of this narrative that quantum computers are going to revolutionize optimization and AI and machine learning and finance, and a lot of that is based on what I would consider to be a foundation of sand. Basically it's based on quantum algorithms that are heuristic, which means none of us can really analyze or prove much of anything about their performance. But in particular, if no one can prove that they won't give an exponential speedup, then people in the business world feel free to make the most optimistic assumption imaginable and say, oh well then, let's just assume that this will get an exponential speedup, because one thing that they learned about 15 years ago is that when you say the word quantum, it was like people's brains shut down. They think it's magic and they're just ready to believe anything. And so I think what I would encourage people to do who are looking at quantum computing startups. The number one question to ask that people don't ask is: but does it beat a classical computer in a fair comparison? Don't tie the classical computer's hands behind its back by using some stupid algorithm. Don't rig the comparison. Look at, actually, the best that you could do with classical heuristic algorithms, such as simulated annealing and methods like that. Compare that against the results from the quantum heuristic algorithm and is there an advantage? And if so, do we expect that advantage to widen as you go to larger and larger problems? Is there any empirical or theoretical reason why you expect better scaling behavior? These are the very first questions that anyone going at it with a scientific mindset would ask, and it is flabbergasting just how much there is where people can raise tens of millions of dollars saying, well, I'm using a quantum computer for vehicle reality and that's enough for people, and they don't ask. Ok, but do you actually beat a classical computer doing the same thing? They have a vehicle routing problem and somehow they never even reach the point of honestly asking that question. And it's sort of by design, because once you do ask that question then there's so little of this stuff that actually survives.

**Craig Smith:** 16:01

Yeah Well, why then would I mean? You mentioned kneeling-based quantum computers for optimization? Why would someone other than research, why would someone advertise that they're using a D-Wave system?

**Scott Aaronson:** 16:23

Well, because it sounds cool and because you know empirically like this has worked, right, Like it's good for D-Wave, you know, like, you know D-Wave gets to, you know, say that people are using our quantum computers to solve these impossible vehicle routing problems. So that sounds great, Right, the company that's doing the vehicle routing. They get to say we're using a quantum computer, you know, the most advanced thing in the world, to do vehicle routing, right, and so everyone is happy, right, you know? And the only problem is well, you know, and strictly speaking, it's not even false, it's just, it's simply irrelevant, right? Because none of it beats using a classical computer for the same task.

**Craig Smith:** 17:05

Right, and it's certainly not cheaper. Right, right, that's right and so.

**Scott Aaronson:** 17:11

But on this optimization problem using D-Wave Anilio-based quantum computer technology, and the truth is, like, you know, I don't even care if it's enormously more expensive, like if you have a proof of principle that, yes, there is better scaling behavior here, right, Then you know we can take advantage of that and you know costs can come down in the future. That's fine. But the real issue with a lot of these quantum annealing approaches is that even theoretically, you know, even supposing that you had a perfect quantum computer that was as big as you wanted it to be, we don't actually know that these approaches would be classical annealing. Yeah, and that's the truth of the matter that I can tell you as a quantum algorithms theorist right, we know that you know these algorithms could eventually get the Grover type speed up, which is a modest, you know, square root speed up, you know. But we don't know that they could get more than that. Hardly ever, or at least not for most of the problems that we care about. And in the near term, you know, we don't even know that they can get any speed up at all.

**Craig Smith:** 18:22

Yeah, so can you describe, but these D-Wave are actually doing the computation for these optimization problems.

**Scott Aaronson:** 18:37

Well, I mean, I presume. So it's like. You know I haven't. You know, for the most part I haven't looked into the exact details of like, like, any time that someone says they're using a quantum computer to do such and such like, you have to think of the quantum computer as the special purpose. You know accelerator, right, this only used for one little subpart of the whole problem, right, it's like no one uses a quantum computer to do you know the user interface, or you know the compiler, the programming language, right? You know we have all of these layers in our abstraction stack, where of course we do and we should use classical computers because they're perfectly adequate, right? And so what people really mean when they talk about these things is that you know you're feeding a classical computer. You know your optimization problem, you know it is. Then you know you have a classical algorithm that's breaking it up into little pieces and then it is trying to find pieces that are, you know, small enough that it can then feed them to the quantum computer, right, and get back. You know answers to those subproblems, right, that then have to be, you know, further, processed classically. Okay, now, that's fine, you know as far as it goes, right, that's how I expect us to interact with quantum computers. You know, going forward, the issue, you know what's always the issue is well, if I did, just did the whole thing entirely with a classical computer, right, if I just, you know, eliminated the part that sends things out to the quantum chip, you know, then would I get a solution that was just as good and just as quickly, right, and I think you know, right now, as far as we can tell, you know, the answer is typically, you know, is yes for basically all the optimization problems that people have looked at.

**Craig Smith:** 20:44

Yeah, and what you were talking about a second ago is a hybrid solution where you just take portions of the problem and give them to you.

**Scott Aaronson:** 20:56

That's right.

**Craig Smith:** 20:56

That's right.

**Scott Aaronson:** 20:58

And then you talk about oh, we're hybridizing quantum and classical computing, as if that's amazing. You know this revolutionary new idea? Well, like, of course you're hybridizing them, right? Of course we use classical computers for everything we can use them for, because you know they already exist and they're so good, right, and we're only going to use quantum computers. You know where they are, where there's some chance that they'll actually give us a benefit, you know, and then you know. Then the main question is do they give us a benefit there?

**Craig Smith:** 21:28

And on and on these, because then very quickly in talking to people they start talking about quantum inspired computing, which I really don't understand. But just on that but before we get there. So you have a D wave and it's got. I don't know how many qubits they're up to in a commercial deal, by the way most people have.

**Scott Aaronson:** 21:53

You know it's amazing how quickly D waves sort of fell off the radar. You know most people's radar, like, I would say, six or seven years ago already, right, and you know, and there's a reason for that, right, it's because you know they, they were like, you know they, they were the earliest ones who realized, like, we can just build systems with hundreds or thousands of qubits and if we do that, then people will be excited, right, and they're not even going to demand to know well, how good are these qubits? Right, you know, how long do they maintain their quantum state? You know, is it actually good enough that I could get any benefit over a classical computer? Right, so they were, you know, in a sense they were pioneers, right, and sort of you know realizing that, that that people just didn't care that much about them. You know the reality of beating a classical computer? Right, but you know, since then, you know now there are, you know, hundreds of companies, you know, I'd say, like of every possible level of seriousness, you know from, just like you know from from companies that are, you know, just some of the best scientists you know that exist working on this, and you know, I really hope that they succeed. You know all the way to like. You know complete frauds who are just like sprinkling the word quantum onto things that have nothing to do with quantum computing.

**Craig Smith:** 23:21

Yeah.

**Scott Aaronson:** 23:22

Right. So you know, but, but in particular, you now have a bunch of companies that can make qubits of enormously higher quality than D-Waves, okay. So, like you know, d-waves have superconducting qubits that maybe maintain their quantum state for a matter of nanoseconds, right? And now you know the like table stakes would be. You know some tens of microseconds, right. So you know, it's just orders of magnitude better, like, if you look at what Google or IBM or Rigeti can do, you know it's a much smaller number of superconducting qubits, but you know. But but but the thing that I care about is that there are a much higher quality, right, there are a quality where, like, it's still not good enough to get a practical benefit over a classical computer, but it is getting there, right, it is, you know, instead of being, like, you know, four or five orders of magnitude away in terms of the error rate. Now, maybe it's only one order of magnitude away, right, and so so, so that, so that that is a genuine cause for excitement, I think. And meanwhile there's been corresponding progress in the other architectures, in trapped ions, in photonics, in neutral atoms, okay, and people have been trying to scale with all of those approaches. So I think that you know that's the.

**Craig Smith:** 24:48

That's the main reason why D-Wave has sort of mostly fallen off the radar at this point, right, right, yeah, you know I've I've done a fair amount of reading, but I still don't understand. So maybe in, in, in the simplest terms you can, you've got this array of qubits, whether they're you know, whatever flavor of qubit they are, they are in superposition. How do you activate them to perform a computation and how do you read the result? I? I, in all my reading. I can't find that answer.

**Scott Aaronson:** 25:38

All right. Well, I mean, of course, it depends on you know at what level you want. You want an answer, yeah?

**Craig Smith:** 25:44

Yeah.

**Scott Aaronson:** 25:46

There's, there's a, you know, there's a mathematical answer, and then there's a physics you know or an engineering answer.

**Craig Smith:** 25:50

The engineering answer I suppose, yeah, okay, okay.

**Scott Aaronson:** 25:55

So basically you know what we're doing, like, like, do you understand? You know the basics. You know what a quantum state is, you know what amplitudes are. What is a unitary transformation Okay.

**Craig Smith:** 26:10

Unitary transformation, I'm not sure All right, all right. Well then, let's, let's, let's let's start there then.

**Scott Aaronson:** 26:16

Okay, so you know the basic ways that you know the basic way that a quantum quantum state can change over time, right? Well, there are two ways, right? One is measurement and the other one is unitary transformation. Okay, you know, measurement is, you know, when you look, of course, and then that's the only part of quantum mechanics where probability enters the picture. Right, when you know, when you make a measurement, you can sort of force your quantum state to collapse, you know, and it makes a choice probabilistically. Right, that's actually the one place where probability ever enters in in any known fundamental physics. Okay, but then, if you leave a quantum system isolated from its environment, then it's changing over time in a way that's actually completely deterministic. Okay, and this is described by the Schrodinger equation. You know, one of the most important equations in all of physics and translated into words, you know, the Schrodinger equation basically just says that isolated quantum states change over time via unitary transformations. Okay, so what is a unitary transformation? So it's so. So, our quantum state is a vector, right, it is a, which basically just means a giant list of numbers right. In this case they're complex numbers, okay, they're called amplitudes, okay, and and like, a central thing that quantum mechanics says is that you need one amplitude for every possible configuration that you could see your system in when you measure it, right? So, for example, if I have a hundred qubits, each one of them could be measured as either zero or one. So that means that I need two to the hundred power amplitudes. I need one amplitude for every possible hundred bit string, okay, so now I have this, this, this vector, this list of two to the hundred complex numbers, and now you know the. In math, the basic way that we change a vector is, you know what's called a linear transformation, right? We just multiply it by a matrix and we get a new vector, right? Okay, and now a unitary transformation is just any linear transformation that always preserves the length of the vector, okay. Why does it have to preserve the length? Because that corresponds to the physical statement that the probabilities of the various outcomes always have to add up to one, right? So you know I always need a vector of unit length, okay, and as long as I have that, then my probabilities will always add up to one, which means that you know, I can interpret them as probabilities at all, okay. So unitary transformations, you, basically you can think of them as rotations or reflections, but not in our three-dimensional space, in this enormously you know, high-dimension, like this two to the hundred power or whatever dimensional space you know, depending on how many qubits I have, right? So? So now you know, if you like, the engineering goal in building a quantum computer is to do, you know, not just any unitary transformations on your qubits but the specific unitary transformations that you want, right, that will, you know, cause the vector of amplitudes to evolve in a particular way. That solves your computational problem, right? And so now you know, you know, you can, just you know, you can, just you know you can, in a classical computer we're trying to build up these extremely complicated operations in order to run the video software for a podcast or anything like that. But inside of the microchip it's ultimately broken down into Boolean logic gates, into these very, very simple operations that we can think of as acting on only one or two bits at a time. The most famous Boolean logic gates are and, or and not. And. Now, a very important theorem in Boolean logic says that if you string together enough and or and not gates, then you can express any Boolean function of any number of bits. So once you have those very simple operations, then they're already universal. Right, then it's all just the question of how many of them do you need. You know, and, and, and and in what order should you apply them? Right Now? I'm telling you all of this because in quantum computing the story is closely analogous, okay, so. So our goal with our quantum computer is to do some giant unitary transformation on all of our qubits, like you know, all thousand or million of them, or whatever. But in order to make that practical, we have to break up the unitary transformation into a product of little, tiny unitary transformations, each of which acts on only one or two of the qubits at a time. Okay, and so these are called quantum gates. Okay, so quantum gates are basically the building blocks of any quantum computation, right? So so, so some of the famous examples of quantum gates. So you know, the not gate that maps zero to one and one to zero, right, that's a classical Boolean logic gate. But that's also a perfectly valid quantum gate, Right, we can have that in a quantum computer, okay, but we can also have gates with no classical counterpart, like, for example, there is a gate that maps a superposition zero plus one to the superposition zero minus one, and vice versa. Right, so, so it is. If it flips, like, like, if the qubit is zero it does nothing, but if the qubit is one, then it flips the sign of the amplitude. Okay, then there's a gate called Hadamard. Okay, so Hadamard is, is, is one of, you know, one of the most popular building blocks, and what? And that's a gate that can take a qubit that wasn't originally in a superposition and put it into superposition. Okay, so the Hadamard gate maps a zero qubit to zero plus one and it maps a one qubit to zero minus one. Okay, that's, you know, you can, you can, you can define it in that way. Uh, and so if I have a bunch of zero qubits and I apply Hadamards to all of them, then I get an equal superposition over all of the possible strings, right, so, okay, but then I also need gates that can create entanglement between multiple qubits. Okay, so a famous example of such a gate is called the controlled knot or the C knot gate. Okay, and what does that do if I have two qubits? Uh, if the first one is zero, then I have two qubits. Then it does nothing. But if the first qubit is one, then it flips the second qubit, okay, so that's why it's called a controlled knot. Right, it's uh, you know, you, you apply, you can either not the second qubit or not, depending on, uh, the state of the first qubit. Okay, and, uh, if I do, if I you know, just using Hadamard and C knot, I can already create an entangled state from a state that was originally not entangled. So, let's say, I have two qubits in the state zero, and now I had a Hadamard, one of them, and now, so now one of them is in the state zero plus one, and now I do a controlled knot from that qubit to the second qubit. So now what I'm actually going to have is well, in the zero branch it's still just zero, zero, but in the one branch, uh, I flipped the second qubit, so now I've got zero, zero plus one one which is an entangled state. Right, so you know, by just putting together these gates in an appropriate pattern, I can create entangled states. You know, even very complicated entangled states. You know, on hundreds or thousands of qubits, what with, uh, what, what I described to you so far is not quite a universal set, but you can throw in just about anything else, like, let's say, you know a gate that rotates a qubit by by some angle, like, uh, you know pi divided by eight. Or you know, you know 20, 22 and a half degrees, something like that, um, uh, and you know, and then you get what we would call a universal set of quantum gates. And what universal means in this context is just that, by a, you know, uh, uh, if, uh, if you apply enough gates from that set, then you can uh affect any unitary transformation on any number of qubits to any desired precision. Okay, so, so, so it turns out that with just a small, a small discrete set of one and two qubit gates, that you already passed the threshold of universality, and that was something that people discovered, uh, more than 30 years ago. Okay, so you know, you could say that you know in quantum algorithms design, you know, what you're always doing is what you're thinking about. Here is my collection of qubits and now here are my allowed gates. And now, which gates do I apply, you know, to which qubits? When you know, and we have a notation for um, uh, for this called quantum circuit notation. You know, it looks almost like a musical score, you know, when you see it written, it was like each qubit is aligned, going from left to right, you know, and then the gates are like different musical notes that we write down, you know, over those lines, okay, and then that, like, like you know, you could say, you know, once they've written the quantum circuit, then that's when the quantum algorithm designers job, you know, or the quantum programmers job, is done. And now this is when the quantum hardware engineers job begins, right, because now what they need to do is take the sequence of gates and translate it into actual pulses that will, you know, act on these physical qubits, you know, in a way that has an effect of doing these gates. Okay, now the good news here, you know, is that you know when we think about, uh, what, what our qubits actually are in different uh, quantum computing architectures. Well, like, in, in, in. In the trapped ion approach, you know, each of our qubits is usually a ytterbium nucleus. Okay, so we have this row of ytterbium nuclei, uh, and, and each one um in co. Usually it, it encodes the qubit uh in its spin state, so like, if it's spinning counterclockwise about some axis, then that represents zero, and if it's spinning clockwise about that axis, then that represents one, right, and it could also be an arbitrary superposition of the two. Okay, so now we have this, this row of nuclei suspended in a magnetic field. And now, what people actually do in the trapped ion approach, you know and this is not speculation anymore, this is all actual experiments that they're, you know that they, they, they actually do. Now they can pick up the individual nuclei and move them around using lasers. Okay, so, they just sort of grab onto a nucleus with a laser, they move it around. And now, when two nuclei get close to each other, right, they have a natural electromagnetic interaction. Right, that causes some, you know, two qubit unitary transformation, right, and now you know, the hardware person's job is just to shape that you know natural unitary into something that looks like a C knot gate or you know whatever was the gate that the uh, the, the quantum algorithms you know designer uh asked for, right, uh, but you know, but, but, but, but, but. But the good news is, uh, like I said, you know there are many, many different choices of one and two qubit gates that will be universal, right, so, so they'll, like, like you know they'll, they'll, they'll all be more or less as good as any other, right, so, uh, so. So now you know, uh, what, what, what they're doing is? They're shaping a pulse sequence, so they're programming. You know the well, uh, a classical computer which is controlling, a microcontroller which is controlling, you know the lasers, okay, to move around. You know the different ions in a particular choreographed pattern. You know that will have the effect of implementing this quantum circuit on the qubits Right, um, and then, at the end, you know there's a different thing that you have to do to measure the qubits. You know you asked about measurements, right?

**Craig Smith:** 39:41

But before we get to the measurements. So when you talk about gates, I mean I, I understand classical computing, the transistors are, are, create gates that are open or closed end and you know that's zero in the one and you build that up, yeah, and in, in, in. In taking this uh, these uh nuclei, for example, are manipulated by what? What is the gate? I mean, is it the, the, the laser, that's.

**Scott Aaronson:** 40:14

I mean, I mean the. The gate is a mathematical abstraction. Okay, a gate is the unitary transformation that is to be affected on this pair of qubits. Right Laser is a way of realizing that gate.

**Craig Smith:** 40:28

That's what I mean.

**Scott Aaronson:** 40:28

Okay, yeah, all right.

**Craig Smith:** 40:30

Okay, and then measurement yeah.

**Scott Aaronson:** 40:35

And then you have to, you know, you have to take your, your, your uh spin state, and now it has to get recorded in something macroscopic, right, and and and typically you know the way that measurement you know devices work, I mean. I mean, you know we have technology, like you know, electron microscopes, that is, you know, that are able to see things at tiny scales, right, but. But? But typically there is a kind of amplification effect that goes on right, where, like you know, you'll have a single, you know photon that, let's say, you know uh deflects in a different way depending on whether you know, you know this. This uh uh nucleus is spinning clockwise or counterclockwise, right, and then that photon could trigger a cascade of additional photons, right, which you know. Then, ultimately, you have a whole photo detector array and it becomes something that's macroscopically detectable, right, so, but, you have to be very careful to only make the measurements when you're ready to right, and in some sense, a large part of the engineering problem of building a quantum computer is to prevent the environment from measuring your qubits before they are ready to be measured.

**Craig Smith:** 41:51

Yeah, yeah, right now, how advanced, how many qubits have they gotten to that you're able to write and read on?

**Scott Aaronson:** 42:05

So you can't just ask about how many qubits, right, you have to ask, like, what can you do with these qubits, how good are they? Because D-Wave will tell you oh yeah, we have thousands of qubits, as many thousands as you want, but then they're just trusting that people are not going to be asking, okay, but can I actually beat a classical computer with them? Or, you know, at any rate, that was the D-Wave of a decade ago. You know, I've heard that they've become more cautious since then. Okay, but for me, you know, the single most important number to look at is actually the two qubit gate fidelity. Okay, which means, like, with what degree of accuracy can I apply a single two qubit? You know, the signaling gate of my choice? Okay. And you know, when I entered the field, you know, 20, 25 years ago, right, it would have been spectacular if you could do a single two qubit gate with 50% fidelity, right, and you know. And then, you know, maybe a decade ago, it became 90% fidelity. And then it became, you know, 95%. And then, you know, with Google's Quantum Supremacy experiment in 2019, right, they had like a thousand gates roughly applied to 53 qubits, you know, in each one with 99.5% fidelity. Okay, now the most recent numbers from the past year that I'm hearing from, you know, both the superconducting and the trapped ion people is like 99.8% or 99.9% fidelity, okay, and this is typically in systems with 50 or 60 qubits. Okay, and you know there are. You know, some people have scaled up, you know, even to. I think IBM has scaled up to hundreds of qubits, but not with fidelity. That's quite that good, right. And now just to help orient people, you know, and where we are right, like the key discovery in the mid-90s, you know, that made people feel like, okay, you know, building a quantum computer is merely a staggeringly hard engineering problem, right, and it doesn't require any new fundamental physics. Was this discovery of quantum error correction?

**Craig Smith:** 44:43

right.

**Scott Aaronson:** 44:44

And what quantum error correction effectively said was that in order to build a reliable quantum computer with as many qubits as you want and as many gates as you want, you know, you don't have to get the error all the way down to zero. Right, like that was the fear. Right, because if you had to get errors all the way down to zero, that's just never going to happen. Right, it's just unrealistic. Okay, but what people discovered in the 90s was that, no, you only have to get the error down to some very, very low, you know non-zero level. Okay, and then there are these very clever quantum generalizations of error correcting codes that can take care of the rest for you and that can push your effective or encoded error rate all the way down to zero. Okay, so you know. So at that, the original estimate from 1996 was, I think, that you needed like 99.9999% fidelity of two qubit gates. So basically, like you could tolerate a one in a million chance of error. Okay, and you know, and that just seemed ridiculously far from where anyone was technologically right, but it at least proved the principle, you know, that there was only a finite amount of engineering that has to be done, right, and not an infinite amount. And, and you know, now, you know, I tell you, you know, with the more recent error correcting codes that people have developed, it looks like you could build a scalable quantum computer with. You know, if you have two qubit gates with 99.99% fidelity, okay, so four nights, let's say okay. And if you remember, you know where I told you, you know we are right now, you know, I think the experimentalists are now closing in on 99.9% fidelity. Okay, on three nights, right? So you know, if you just plot that on a graph, like it looks like it's pretty on track, right, it looks like, yeah, you know, if, if, if this effort continues, then you know we should eventually, you know, get to the threshold where you know, after which error correction becomes a net win. And now, once you've crossed that threshold, then how many qubits you want is just a question of cost, right, it's just just like with a classical computer, right it's? You know, it's like, okay, if you want, you know, you know two million qubits instead of one million, then you know, then that's more engineering that you have to do. But you know, you have figured out how to keep your qubits alive for as long as you need them to be, and then there's no fundamental limitation on how many more you can add. Okay, so that's so. That's the real goal. To get to this fault tolerance threshold right, and you know the things that people can do now with 50 or 60 qubits, you know they're. They're very cool as demonstrations, you know, and they're important as proofs of principle, right, but none of this yet has been error corrected, right, and that and and and. That's the real goal.

**Craig Smith:** 47:57

Right, right, and, and none of it is. I mean certainly the quantum supremacy experiment, but none of it is beating a classical computer.

**Scott Aaronson:** 48:09

Yeah, well, I mean, I mean the quantum supremacy experiment. You know, arguably, you know, sort of is right, you know, not, not for a useful problem, you know, but but but at least for some contrived benchmark. You know, and and you know that came out of work that my student and I did in 2011, where we said, you know, hey, you know it's not. You know people are talking about like, like what, what can you do? That's useful with a non-error corrected quantum computer. But, you know, maybe we should just first try to pin down. Can you do anything at all? That is classically hard, right, even if it's not useful. And we came up with a proposal for how you might do that which was called Boson sampling, which was adapted to, to, to optical quantum computing, right, and you know. And then you know the optics, people got very interested and started doing very, you know, small scale experiments with, you know, five or six photons, right, at that point, it's still trivial for a classical computer to simulate. But then what happened was that Google, in 2014, I think, hired John Martinez, who was one of the leading superconducting experimentalists, and Martinez actually said, like, let's go for this, let's just do a quantum supremacy demonstration. You know, with, with, with, you know, I think you know they were hoping for 60 or 70 qubits, okay, and, and you know, this wouldn't exactly be Boson sampling, because, you know, their, their, their, their hardware was not designed for it. They said, you know, let's do something like Boson sampling, but adapted to our hardware, so you know. So we talked to them at the time and we helped, helped them figure out what that experiment would be, you know, and then we sort of adapted the theory of Boson sampling to the kind of hardware that they had. And then, you know, in 2019, they actually announced results that, with 53 qubits and you know, about 20 layers of gates, about a thousand gates and all, they solved this, this sampling task that you know. At the time they estimated that, you know, maybe it would take 10,000 years for a classical computer to do the same thing, and then that number got quoted all over the press. Unfortunately, you know, you have to be really careful with these numbers, right? Because then what happened was that the classical computer scientists came along and said, no, we can actually, you know, spoof this a lot faster with a classical computer, right, and now they can, I would say, if you're willing to spend enough money to just have enough parallel classical processors, then you can actually simulate the Google experiment faster than the experiment itself. So it all just comes down to energy and money. And by those metrics, I would say, on these quantum supremacy benchmarks, the quantum computers are still winning, but only by a couple of orders of magnitude, not by a whole lot. So there's an urgent need for better quantum supremacy experiments that will reestablish a clear speedup. I think, given the way that the technology has advanced just within the last four years or so, that should definitely be possible. But in the meantime a lot of the companies have said they don't really care about quantum supremacy anymore. They just want to go straight for error correction.

**Craig Smith:** 51:45

Right, right, and so error correction is kind of the.

**Scott Aaronson:** 51:49

That's the holy grail.

**Craig Smith:** 51:51

Yeah, the foundation or the frontier across which will be into practical quantum computing.

**Scott Aaronson:** 51:58

Yeah, I mean, again, there are some people who hope that we will get lucky and be able to do practical things even before fault tolerance. Okay, there's a term people use for this NISC, coined by John Preskell: noisy, intermediate scale quantum computing, mm-hmm. You know I mean by analogy. Right before the transistor was invented, we had only much noisier building blocks for classical computers, like vacuum tubes or electromechanical relays, but you know, they still worked well enough for people to do various useful things with them, among which was winning World War II, right, so you know, so you could hope that maybe the same will happen here that even before we get truly error corrected qubits, you know we'll have qubits that last just long enough that we can eke out some advantage. And I think you know, if we're going to do that, then the best hope by far is that we'll do that for some sorts of quantum simulation problems, such as simulating the properties of various materials, okay, and there's a much better hope of doing something that is scientifically interesting, you know, and new and classically hard, than there is of also doing something that's commercially useful, which is kind of a higher bar to clear. And people are working on that, and so you know. But the point is, you know there is no guarantee right. Like you know, nature will have to be kind to us, or, you know, for us to get clear quantum computing speed ups before error correction. You know, after error correction, then we're much, much more confident that we could get such speed ups.

**Craig Smith:** 53:51

Yeah, yeah. Well, we're almost up to an hour. Yeah, can you just give a synopsis of your safety work at OpenAI? You did a very good job on the collective intelligence webinar.

**Scott Aaronson:** 54:06

All right. Well, so OpenAI approached me a year and a half ago because you know several people there were fans of my blog and they said you know, would you take a leave of absence? And you know, work with you know, work for us on what computational complexity theory can do for AI safety. You know and my first reaction was you know, why do you want me? I'm a quantum computing person, right, you know. We know what you know, I mean I've been, you know, and already at the time I was, you know I had seen GPT-3, you know, I was bowled over by it, you know, I knew that it was this, you know, unbelievable advance from the previous state of the art in chatbots, right, but I was like you know. First of all, you know, the progress in machine learning, you know, has been driven almost entirely empirically, right, it's. You know, for the most part it hasn't depended on theory, right, it's depended on, you know, having lots of data and lots of computing power, right, and you know. And then tweaking the algorithms until you find the ones that just empirically turn out to work well, right, and you know, no one really understands, you know, at a deep mathematical level, why any of this works. Right, they just, you know, they just train the models, and then they try them out, and you know, and it turns out that when they're big enough, they do work. Right, so you know. And then, furthermore, you know, none of it has anything to do with quantum computing, you know, except that, okay, you know, machine learning and quantum computing both involve extremely high-dimensional vectors, right, you know, so, you know, but beyond that, you know, they're just very different fields. But you know, they made a case to me that well, no, you know, like the problem going forward is not just going to be how do we get this to work, but how do we ensure that it's safe? Right, and that does seem like a problem where we have to go back to first principles, where we have to think about it theoretically. You know, and they gave examples of, you know, some of the big results that we know in computational complexity, about how, like, a weak verifier can check the behavior of a super powerful but untrustworthy proofer. And they said, well, you know, that's basically the situation that we're in. Right, you know, the super powerful proofer is the AI and you know the limited verifier is humanity. Right, and you know, and maybe you can port these results over. And you know, and I said, well, you know, it sounds like, at any rate, I'd be able to find something interesting to work on. But you know, so you know, I'm teaching my quantum computing courses, I'm supervising my students, you know, I have a whole you know, so maybe some future year I get involved. And they said well, you know, trust us, this is going to be a really big year for AI. You want to get involved this year? I said, you know, and, and, but you know, they were nice enough to let me do it from Austin, where my family is and, just, you know, travel to San Francisco periodically. So you know they made it very hard to turn down, and so I. So, I did, I started working for them and you know, I found, you know, so it's now been a little over a year since I started and you know I found various projects to work on. Maybe the easiest one to explain is a water market. Okay, so you know, like I've not been able to, you know, predict where all of this is going. You know, even, like you know decades in the future, as like people constantly ask me to do now. But I'm proud that at least I was able to see about three months into the future, right? So, before chat GPT was released, you know, I just had this moment of terror when I thought, oh my God, like every student in the world is going to be using this to cheat on their homework, aren't they Right? Or at least is going to be tempted to write, and you know, and, and, and every troll is going to, you know, have this amazing tool for spamming every forum on the internet, right, or impersonating people, or you know you know what? about propaganda campaigns or fraud, right, and for all these different categories of misuse, you know, like, like they would all be, you know, much harder if only we had a way to detect, you know, which text was generated by large language models and which was not, so, so. So then you know, I think you know, I think you know, I think you know, I think you know, I realized that that that this attribute, you know, this provenance or attribution problem was going to become really central, right, and you know there are a bunch of different approaches that you could imagine to it, but one, one particular approach that I've worked on is called watermarking. Okay, and this is where we slightly changed the way that our language model, you know, such as GPT, operates. So language models are inherently probabilistic right. You know they're always like that. You know you have this transformer, you know the neural network that's always taking his input. You know the context of, like, the previous, however many words, and then it's generating a probability distribution over the next word, right, and you know, now, now, like in normal operation, all that would happen next is that we sample according to that distribution, right, sometimes the distribution will be very, very centered on one possibility, Like if I say the ball rolls down the, you know, then GPT is nearly certain that the next word is hill, right, but you know, in other cases it's balanced between several possibilities, right. Or you know even dozens of them, okay, so, so, so. So, with watermarking, what we do is we choose the next word or the next token, you know, in a way that is secretly deterministic, okay, but which looks random, right, it looks like we're doing it according to the distribution that that GPT said to sample from. You know, casual users can't tell the difference at all. But we're actually choosing the next word deterministically, in a way that is biasing a score that we can calculate later, given only the text, right. So we're sort of doing it in a way that systematically favors certain combinations of words over others, right, you know combinations that would be otherwise arbitrary and you know meaningless. Okay, and then you know, if later some teacher gets a term paper and they suspect that it was written by, you know, using chat, gpt, they would be able to submit it to a detection tool and that detection tool would calculate that score, you know, using the key of the pseudo random function, you know which it would have. And then you know, if that score is really really unusually large, then you would be. You know, depending on how many, on how long the document was, you could be statistically almost certain that, yes, gpt must have been involved in writing this.

**Craig Smith:** 1:01:51

Right, and would that work if only portions of the document were?

**Scott Aaronson:** 1:01:58

Yeah, that's an excellent question. So I do have an algorithm also to do that, to take a long document, portions of which were, you know, we believe were written by GPT, and then identify which were the most likely regions to have been GPT generated. Okay, now, none of these approaches are foolproof. Okay, you know, one can think of a tax, you know that would get around them, some of which are, you know, like imagine a student who asks GPT to write their term paper, but in French, and then they put it into Google Translate, right, and that would remove any watermark that we currently know how to add Right. And so you know how you watermark at the level of the ideas, you know, at the semantic level, like out of way, in a way that would survive. You know all these sorts of you know rephrasing, sort of. You know you know mere changes to the, you know local changes to Syntax and things like that. I think that's a very, very hard question.

**Craig Smith:** 1:02:58

But that's it.

**Scott Aaronson:** 1:02:59

That's fascinating, and you're optimistic about the well, I mean, I mean, I'm, I'm, I'm optimistic that we can deploy things. You know that will, you know that that will help at least somewhat and especially that we'll be able to learn more by deploying them. Right, yeah, you know, I'm optimistic that AI safety, you know, after decades of armchair speculation about it, is finally an empirical subject. And you know, I think it might be very difficult to stay one step ahead of all of the misuses that people come up with. Right, it might be a cat and mouse game, but you know, in other cases, you know, people have just bitten the bullet, right, I mean. I mean, you know, we have a giant, thriving software industry. You know, despite, you know, the ease of pirating software. Right, we have, you know, search engines like Google. You know, despite all of the attempts to to game their results, right, so you know, so it might just be that that you know it will take a large effort to stay ahead of the misuses, but you know, at least in the, in the near to medium term, yeah, I'm reasonably optimistic that that we, we can, you know, make progress. We can break down the problems into subproblems and tackle them. Now, once you have AI that is just Better than humans at everything, right? Or you know, let's say, ai that you know is the humans as we are to orangutans, right. Then you know, I don't know what kind of world that leads to, right? I don't know how to ensure that that world is safe.

**Craig Smith:** 1:04:43

Yeah, but you know me.

**Scott Aaronson:** 1:04:45

My defense is I don't think anyone else knows either.

**Craig Smith:** 1:04:47

Yeah, yeah and tonight I have one last question. All right, the, the. I've been talking to people about the pros and cons of open sourcing, generative AI, yes, and there is an argument that it's great because you know it expands the use cases, and, and you, you get this network effect of people improving. And then the other side who says, well, really, it's such an asset, yeah, and so Game, that open source really doesn't have the resources to compete. But what I'm interested in is, even at the lower, with, with smaller models, that once they're open source I mean things like this watermarking, yeah, it's not gonna. That will Work on a proprietary, yeah, but once it's open source, it's anybody's game. Is that right?

**Scott Aaronson:** 1:05:52

No, you're, you're absolutely right, and that's true not just for watermarking but for pretty much any safety mitigation that you could possibly think of. Right, yeah, once the model is open source, then people can do what they want with it. And you know, and empirically, like you know, you know, if you, if you fine-tune your model via, like this, rlhf, reinforcement learning with human feedback, you know to be, you know to only give, you know safe, you know helpful in offensive answers, right, it takes about two days for people to take an open source model and remove that fine-tuning. Yeah, you know the model will spell, you know, whatever racist, infective or bomb-making instructions you want it to, right, and so, so that, so that that that's the trade-off with open sourcing models, right, and so I think it fundamentally depends on how powerful the model is. Right, some stuff that's at the level of GPT-2, you know, has already been open sourced, right, you know. And and stuff at the level of GPT-3, you know, probably, you know, you know is in the process, right, or you know, will, you know, either is or soon will be right, gpt-4, you know, you know, I feel that you know open AI made a very defensible decision to not open source that right, because you know there there is probably, you know, some some mayhem that that people could already do with it. You know, if they had access to the weights, I mean, you know, even just just aside from you know the obvious commercial reasons why open, yeah, you know, doesn't want to open source it, right, I think. I do think there's also a safety issue there, right, and eventually, you know you may get eyes that are powerful enough that open sourcing them is kind of like open sourcing thermonuclear weapons. Right, it's, you know, it's just a thing that you don't want to do right, yeah, and?

**Craig Smith:** 1:07:58

This episode is sponsored by Crusoe Cloud. High performance cloud computing or low environmental impact? Crusoe Cloud was built because the innovations of the future need both. Crusoe Cloud is a scalable, clean, high performance cloud, optimized for AI and HPC workloads and powered by wasted, stranded or clean energy. Crusoe offers virtualized compute and storage solutions for a range of applications, including generative AI, computational biology and rendering. Visit crusoecloud.com that's C-R-U-S-O-E C-L-O-U-D dot C-O-M to see what climate-aligned computing can do for your business. That's it for this week's episode. I want to thank Scott for his time. If you want to read a transcript of this conversation, you can find one on our website. Eye on AI that's EYE-OM Dot AI. And, as always, the singularity may not be near, but AI is about to change your world, so pay attention.