SDS PODCAST EPISODE 1041: CONNECTION IS NOT COGNITION: WHY MULTI-AGENT SYSTEMS SILENTLY FAIL, WITH DR. VIJOY PANDEY

Jon Krohn:	00:00:00	Your AI agents can pass messages to each other flawlessly and still completely misunderstand one another. Today's guest has the data to prove it and an open source fix. Welcome to another episode of the Super Data Science Podcast. I'm your host, Jon Krohn. My returning guest today is Dr. Vijoy Pandey, SVP and general manager, in other words, the head of Outshift, which is Cisco's incubation engine. On his third visit to the show, Vijoy reveals that every multi-agent session his team analyzed contained at least one hidden cognitive failure, and he explains the new open source protocols Outshift built to keep agents aligned. He also details a record-breaking quantum network running on real New York City fiber and introduces new chips being built to compute on probabilities instead of zeros and ones. Enjoy. This episode of Super Data Science is made possible by Anthropic, Dell, Nvidia and Parallel.
	00:00:59	Vijoy, welcome back to the SuperDataScience podcast. Such a treat to have you here. How you been doing?
Vijoy Pandey:	00:01:06	Well, it's been a treat for the past two times and it's a treat again to be here. So thank you for having
Jon Krohn:	00:01:11	Me. For sure. Two of my favorite ever on-air conversations. So delighted to have you back. We understand that it's a fast moving space and so you're going to have lots of updates for us. Right now, everyone seems to be trying to make individual agents better. That's a lot of what we hear about. Thats a lot of what we talk about on the show. But you, since you were last on the show nine months ago, you've been putting lots of multi-agent systems into production. What have you learned from those deployments?
Vijoy Pandey:	00:01:46	Let's start with where the agentic computing arc is, Jon. So if you think about agentic computing, it started with isolated agents. That's what you just mentioned. I think we went from isolated agents to islands of agents where

you can have many agents, sub-agents all working together, but within the same harness. So within the cloud code or within codex or within the harness of your choice. What we did was we took isolated or islands of agents and connected them. So we moved from isolated islands to connected agents and we've been deploying that over the past 18 months plus in a variety of verticals. So we have built multi-agent systems to solve for pain points and use cases in digital surveillance, in network configuration, in cloud platform engineering, in medical health and diagnostics. So a bunch of these use cases. And what we found is the need is massive first and foremost.

00:02:55 We are looking at agents that come from different vendors that sit on different clouds that can access different tools with different identity and auth, and they belong to different organizations. So the need for connecting these multiple agents is actually massive. We've also seen a whole open source ecosystem appear along the way. So we ourselves found agency and this was an open source ecosystem that's spelled A-G-N-T-C-Y, by the way. And this open source ecosystem was created for agent discovery, identity, messaging and observability. Since then, we've actually worked with A2A and MCP under the AIF. This is the Agentic AI Foundation. And we've also formed some other open source projects like AI catalog with the Googles and OpenAIs of the world. So all of this has been happening to then connect agents. What we found out was though as we started connecting agents, we realized that agents are not coordinated.

00:04:01 So just because you connected agents doesn't mean that they're coordinated agents. And that is leading to a whole set of failures that we are calling silent failures with loud implications because you can't measure them. You see the dashboards being green today, but the failures are there and your multi-agent systems are producing wrong outcomes.

Jon Krohn: 00:04:27 Yeah, this reminds me, Vijoy, of a conversation that we had on the show a few weeks ago. So in episode 1030, I had Garav Patak on the show and Gaurav was talking about how when humans disagree about some figure, there's some important figure that you're going to be presenting to executive leadership. And so people in a meeting, three different people come with three different opinions of what this figure should be. And you'll have an argument and you'll figure it out and you'll come out of the meeting with either the answer of, okay, this was actually the right number to work with, or you come out with some action plan to figure out what the right figure is. But when you have three agents that come to some kind of virtual conversation in some embedding space with their figures, it can be a completely silent problem.

00:05:28 Like you're saying, they can just... The process that unfolds, you probably have a lot more insight into this than I do, but it seems like we can just very frequently have that process, which should have been argumentative, just result in one figure being presented very confidently as opposed to trying to figure out what's really going on.

Vijoy Pandey: 00:05:50 So first of all, Jon, thank you for bringing that up because I think first and foremost, I need to go and talk to Gaurav and we need to start collaborating together. But one of the things that we've found in all of these vertical segments that we've deployed multi-agent systems in is exactly that, which is agents together can connect, but they cannot think together. And the other way to thinking about this is connection is not cognition. So there are a whole set of cognitive failures that happen when agents start exchanging meaning, when they start exchanging semantic information. So I'll give you some other examples. Let's take a very simple example of an SRE agent and a SUE agent. These are things that we deal with all the time. When one of the agents says, "Let's roll back," and SRE agent might mean the last

configuration. For the SUE agent, configs are not really the thing that they deal with on a day-to-day basis.

00:06:54 So they might think of let's roll back the entire software version. And so there's the keyword rollback means different things to different agents because that's what they're accustomed to. And we call this a grounding problem. And take something even bigger, like a protein folding specialist trying to propose a candidate, but a specialist proposing a protein candidate is not the end all and the be-all. You have to take it to a wet lab agent that then runs the synthesis, an agent that tests for efficacy. Then you have coordination with a regulatory agent that is looking at what FDA can approve and what that proposal looks like or the pathway looks like. You can then take it to a supply chain agent to see whether it's manufacturable. That happens all the time. Take an enterprise that we are so familiar with. You have a two million deal that is flowing through a deal desk flow and you have sales and legal and finance and product and marketing and all of these people mean different things when they say Sweeze will say the code is ready.

00:08:04 That does not mean you can sign it up to the customer. And that does not mean that sales should be communicating that date to the customer. So what does ready mean is actually pretty fluid and ambiguous in these kinds of situations. So we've been looking at all of these cognitive failures and what we've been realizing is the system, an agentic system can become more consistent over time through these cognitive failures without being more correct over time. And these failures tend to compound. And of course we've looked at multi-agent systems in all of these vertical spaces that we've deployed them in, but we also took public traces from chat dev sessions, from MMLU probes that exist out there and we analyze them. And what we found was pretty surprising. We found that 100% of all multi-agent sessions had at least one cognitive failure.

Jon Krohn: 00:09:08 Oh wow.

Vijoy Pandey: 00:09:09 We found that these failures compounded. So if there was one failure, there was still 98% accuracy in the multi-agent system output. You go to two failures and it drops to 83%. When you go to three failures, it drops to 16% accuracy.

Jon Krohn: 00:09:27 16, one six.

Vijoy Pandey: 00:09:28 One six. That's as good as randomly picking an outcome. And so these failures that again, you don't see them because your observability dashboards are green, but these are silent failures that are going to impact the outcome that you're looking for from a multi-agent system.

Jon Krohn: 00:09:47 I want to hear more about these tests and how you evaluated them in a second. I want to clarify quickly for our listeners what SWE and SRE is, probably most listeners are aware, but so SWE is software engineer. It's a TLA, a three letter acronym for software engineer. And then SRE is S-R-E, site reliability engineer. And the key point that Vijoy is making there is that to those two different kinds of personas, a software engineer and a site reliability engineer, if you think about those as human people, then those two different people would have different ideas in mind of what the word rollback means. And so similarly, if you create an agent that is specialized to be a software engineer, a SWE or a site reliability engineer, a SRE, the context that you provide them with, the expectations for them would be different for what rollover means for those agents as well.

00:10:44 And I think one of the key things here is if you had a software engineer and a site reliability engineer as humans, kind of similar to my example earlier about three people disagreeing over a figure, you are likely if you had the SWE and the SRE humans in a meeting and they

started talking about rollback, one of them would probably say, especially a more experienced SWE or SRE would say, "Wait a second, I think we might be talking about different kinds of rollback here. Let's make sure that we're on the same page and we have the definition right." And I think what has happened in the quick evolution of multi-agent systems is that we're not getting the place for those cognitive clarifications to happen and so cognitive failures occur.

Vijoy Pandey: 00:11:30 That is correct. And I think the way to think about this is we've connected agents. So we've built the infrastructure where agents can now talk to each other and pass perfectly valid, accurate messages, but they don't align on what those messages mean. And that is the next step in the evolution. So yes, things are happening quickly, but it also means that there is a lot more work to be done. It's not that we throw away the stack that exists. We need to build upon the stack that exists. So we had isolated agents, we built the stack to now connect agents. Now we need to build a stack that can exchange meaning that can exchange cognition state. So it's a perfectly fine point in that evolutionary journey for agents as well.

Jon Krohn: 00:12:15 Yeah. It's the only way we can go. It's Terra Incognita for us all as we figure out how to do this. Tell us a bit more about the evaluation. So when you were seeing those situations where when there's three cognitive failures in understanding between different agents, you're getting that 16% accuracy. How do you conduct those tests? I guess that's something that you're doing at agency. Again, that's A-G-N-T-C-Y.

Vijoy Pandey: 00:12:43 Right. So right now agency and 82A and MCP and the entire ecosystem that exists out there has been built to connect agents. And when you connect agents, yes, we do have a whole observability stack that we actually worked with Microsoft and pushed it to an OpenTelemetry about October of last year, so a year ago, roughly. That

OpenTelemetry observability stack actually tells you whether agents are working well, whether their uptime is okay, whether the reasoning engines are functioning properly. What it does not tell you is what happens when these two agents start exchanging messages and when they start exchanging meaning. So you can still bring in connected agents that can have different goals, that can have different context and assumptions and constraints and beliefs. And that was some of the examples that we saw earlier. And when you bring these agents together into a multi-agent system, what you have is a lack of grounding.

00:13:45 What you have is a lack of alignment on terminology. You need to discover terms that you don't understand. You need to resolve conflict that you disagree upon. Then once you have aligned on intent, the next step is to then coordinate just like human teams. So once we've aligned on the goal that is let's go ahead and solve this problem together. Let's build this drug at this percent efficacy to pass FDA scrutiny and then be manufacturable, let's manufacture that drug together. So that's the goal. Once you start working on that problem, just like human teams, you need to coordinate and negotiate because things are not going to be perfect. And every agent comes in with their own local optimum. Every agent comes in with their own local incentives. So when you have this coordination and negotiation problem, every agent needs to compromise a bit and then work towards a global optimum.

00:14:47 So that's the coordination and negotiation problem. That's the second step. And then finally, once you've achieved a goal, is there a verification layer? Again, at the cognition semantic layer, is there a verification that tells us that you've done your job well? It's almost like all of us going through a performance review at the end of a financial year and getting a score as to how we've done. So we need to do that again at the semantic knowledge

cognition layer. So those are the three steps, the shared intent, grounding, resolution, discovery, then the coordination and negotiation to actually go through that task and achieve that intent. And finally, the verification step to ensure that we've met those goals at the cognition and semantic layer. So that's the stack that we're building out now. We are calling it the internet of cognition, and it's something amazing because it actually follows what's happened in human intelligence right to the dot, where human intelligence also started out through individual intelligence getting better over hundreds of thousands of years.

00:16:05 Then roughly 70,000 years ago, there was the invention of language which caused a cognitive revolution. And cognitive revolution pivoted us from individual intelligence to collective intelligence. And that's exactly where we are with agentic workflows where agents are getting bigger and smarter over time. And we see that all the time, but they're still isolated. They're just now getting connected through the work through agency and A2A and MCP. And now we are building the internet of cognition to move towards collective intelligence where we are inventing protocols, where we are inventing shared knowledge, where we are inventing ways for agents to then share reasoning together.

Jon Krohn: 00:16:52 Yeah. So we're trying to have the same kind of explosion in collective intelligence facilitated by a common language. Now with agents like we got with humans 70,000 years ago, it seems to me like one of the special things about finding that common language in the way that you are doing it at agency is that, correct me if I'm wrong, you're trying to make sure that that language really is common across whatever kind of agentic framework is being brought to the table.

Vijoy Pandey: 00:17:24 So first and foremost, agency is an open source stack along with A2A and MCP that allows agents to connect

together. And the good thing that has happened over the past, I would say nine months since we last talked is that there is actually convergence across all of these open source projects that have popped up to solve for the connectivity layer, which is a really hard thing in the industry as you can imagine. Convergence between humans is still a hard problem. And so the fact that agency and A2A and MCP, they all sit within the Linux Foundation and they're all working together on a common infrastructure stack so that we can discover agents in a common way. We can describe agents in a common way, that's AI catalog work. Then we can discover agents in a common way. We can then attribute identity and access controls to agents in a common way and then communicate whether it's agents to tool using MCP or agent to agent using A2A, using a common messaging layer, which we call SLIM or secure low latency messaging, interactive messaging, that stack is becoming common and integrated.

00:18:49 So that's a good thing first and foremost. But what's still missing is that cognition layer. What's still missing is that semantic layer. And what we've realized is that the entirety of the networking infrastructure for the past 40 plus years was built for connecting data endpoints. And now we have a new kind of endpoints that is starting to think. These are intelligence endpoints, and so we need a new layer of protocols that connect intelligence endpoints that looks at meaning and not just data as a blob. And that's what we are building. We published a paper around this late last year around online protocols, but we are actually building those protocols right

Jon Krohn: 00:19:35 Now. Excellent. Well, that all sounds exciting, Vijoy, but how much of this is really shipping and something that my listeners can use as opposed to being another white paper?

Vijoy Pandey: 00:19:48 Well, we just released an entire stack that allows you to look at first silent failures in your multi-agent systems, to measure them, to characterize them as to what kind of a failure is it? Is it goal alignment? Is it coordination and negotiation? Is it a verification problem? So one of those three. And then we've released two protocol implementations to solve for two of those problems. So let me walk you through that. So first and foremost, think of this entire architecture in three layers. So at a very abstract level. First and foremost, there are these protocol layers that get triggered when you see something going wrong.

00:20:35 It's like you're driving a car and you have lane correction turned on. So when you're driving a car, you're all good, you're running down the highway, no problems. But then when the car starts seeing that you are digressing, you're moving into the other lane, maybe you're sleeping a little bit or whatever it is. I don't know. I don't recommend doing that. Pull over and sleep. But there might be something, maybe you're chatting with your kids or they're creating a ruckus or the car starts veering into the next lane or maybe there's construction on the road, who knows? So when it starts veering into the next lane, your car then triggers this feedback loop to you saying, "Get back within the lane." We think of this entire protocol layer in the same way. So there is an observability and explainability platform called OXP, observability and explainability platform.

00:21:32 What it's doing is it's measuring all of the... It's looking at all of the traces flowing in your multi-agent system and it's doing this through the open telemetry extensions that we've already shipped last year that I talked about. So it's sitting in agency, it's sitting in hotel, you can use it, it's defacto standard, open source, and it uses those traces in this multi-agent system to constantly look at what's happening at the semantic layer. And it's looking at cognition drift. So whether it's alignment drift, whether

it's coordination drift, whether it's verification drift. So any one of those failures that we talked about, it's looking for signals of that. So first and foremost, you can just use OXP and measure cognition failures in your environment. That's the simplest thing you can do. And you will see similar results that we saw. And so characterize them, look at what's happening.

00:22:34 But then what we do is this OXP platform, as it measures drift, it then triggers some protocols to fix that drift, just like the lane change. So when it sees intent drift, before agents even start working on a problem, it can detect drift in intent that these guys are not even aligned in what they're supposed to do. We trigger a protocol called ACORD. It's an acronym. I'm not going to say what it means. I'm sure I'm going to butcher it up, but it's an acronym that actually allows for intent to be aligned. And this is a four phase protocol that produces a locked contract for your intent. It's a shared artifact that all agents then commit to before any work can start. So that's the first protocol that we can trigger and that's available right now.

Jon Krohn: 00:23:33 And that ACORD protocol is spelled A-C-C-O-R-D?

Vijoy Pandey: 00:23:37 That's correct. And it's A-C-C-O-R-D, and it's an acronym and you can find it and you can read the acronym yourself, but it's basically locking the intent and it's a shared artifact across all of these agents pre-work. So that's the first one. Then once work starts, again, your OXP platform is looking at all of these traces and what's happening between these agents and the semantic layer. And if you see consensus drift, or if you see coordination getting drift or getting in failure board, then we trigger another protocol and that's CONCORD and that's C-O-N-C-O-R-D. That's also an acronym. And what it does, it's a consensus protocol. So it resolves decision conflicts or it resolves divergence during a live task, and again, brings the agents back into the lane. So that's the

Concord protocol. So these are the two protocols we've actually released so far. You can imagine us releasing other protocols that look at other kinds of cognitive failures in the future.

00:24:48 And the one thing to keep in mind again is these get triggered only when you steer across your lane. In regular traffic, you're doing fine. You're just running using standard GRPC or MCP or A2A, the protocol of your choice. That's what you're using, that's how agents are communicating. We don't touch them. We only measure them. And when we see these drifts and failures, we trigger these protocols. They bring you back in line and then they get out of the way as

Jon Krohn: 00:25:20 Well. I love it. So it doesn't matter how I create my agent or how me and a bunch of different teams create different agents using different frameworks for each of those different agents. And then we can bring them together. We can coordinate together on getting the right intent with Accord and then making sure that all the agents are staying aligned with Concord. And all of this is part of what you call OXP, the observability and explainability platform. Did I kind of summarize that correctly?

Vijoy Pandey: 00:25:53 All of these are actually triggered by the observability and explainability platform.

00:25:57 I see. So the platform itself is almost like sitting on the side looking at what you're doing. It's like your auto navigation in your car is looking at what you're doing. It's actually not driving the car, but as you see the car do things it should not be doing, steering into a different lane or getting too close to the car in front of you, it then triggers a protocol to then react and get you to safety or within your swim lane and then you're back to driving the car. So that's the way to think about this. And at the bare minimum layer, you can use OXP to just characterize all of these cognition failures in your multi-agent systems.

So even if you don't want to, I don't know why you wouldn't want to align them as a next step, but even if you don't want to align them, you can just observe and measure and characterize your multi-agent system at the cognition layer.

00:26:51 So that's already pretty useful. And by the way, one more thing on how you can use them because it is a pretty heterogeneous world out there and there are agents that are coming in from different harnesses that do sit on different clouds. And so the heterogeneity in this environment is pretty massive. So we've tried to make usage of OXP as well as these protocols pretty straightforward. So OXP, like I described, is based on OpenTelemetry, which is the defacto telemetry stack for any cloud native application. So if you're using any cloud native stack, Kubernetes, whatever, you will get OXP as an extension to that period because we are looking at OTEL traces and we are building the semantic clear ourselves just by looking at the OTEL traces so you don't need anything extra. So OXP is by default through a cloud native stack. The protocols themselves are available through what you prefer.

00:27:59 So you can use a skill and deploy that skill in your agent, which is the least friction way of using these protocols. And they again, sit on the sidelines till they get triggered. You can use a sidecar based approach. So this is what we've done for a decade now, again, in a cloud native stack. So an Envoy based Sidecar as an example, or an EBPF based sidecar. So there's a Sidecar based approach, but we also have bindings to some of these common lower layer protocols. So there are bindings for A2A, they're bindings for MCP, they're bindings for Slim. This is an agency messaging protocol. So you can actually use a cord and concord above MCP, above A2A, above SLIP. So pick your approach. We give you the least friction way of using Accord and Concord, and we'll be doing that for all of the protocols that we ship.

Jon Krohn: 00:28:58 Amazing. And I am assuming based on the way that most of the things that you and I have talked about in past episodes on this show, that all of this is an open standard. It's freely available for anyone, right?

Vijoy Pandey: 00:29:10 That's great that we brought it up. All of this is open source. So not just the open telemetry traces and extensions, which are open source last year, A2A, MCP, Slim, these things are part of the ecosystem today for connectivity, but Concord, as well as the engines that support measuring cognitive drift are all open source. So please use them, please characterize your multi-agent systems, give us feedback, get involved in the community. That's what we ask.

Jon Krohn: 00:29:43 And why is Cisco doing this? Why is Cisco doing so much, putting so much R&D time into developing these free open source opportunities, these free open source technologies?

Vijoy Pandey: 00:29:56 First and foremost, we are outshift. We are the incubation... An engine for Cisco. Our job is to look at where technology is emerging and then figure out how we can build products and projects in those emerging technology spaces, reduce tech risk and market risk for Cisco as Cisco tries to enter those market spaces. So that's what we do. And a big chunk of what we end up doing, because we are always exploring the frontier of technology, a large chunk of what we do is educate the market. A large chunk of what we do is figure out what makes sense. So exploration is a big deal for us. And then as we explore products and projects in these spaces, we can figure out which problems matter to our customers so that when we do build products, yes, we are not a charity at Cisco, we will be building businesses and products in these spaces, but we are gathering early signals and figuring out which problem spaces and which technology stacks actually matter for our customers, for the use cases that they're interested in.

00:31:14 So we are so early that we are exploring, but soon we'll narrow it down. And yes, there will be products that Cisco will build that incorporate a lot of what we are doing. But also one of the things that we believe in is that open source is actually the right approach for tackling any technology problem. We are big believers in open source because open source is fundamentally advantages to every player in the ecosystem. Whether it's a vendor, whether it's a developer, an operator, whoever it happens to be, a cloud provider, an open source approach is fundamentally more advantages and more profitable than having closed ecosystems. And we are firm believers in that. And we've always relied on open protocols, open standards, open source to then build products and projects in those spaces.

Jon Krohn: 00:32:18 I agree with you a hundred percent. I also am a big proponent of open source, not only as something that's great for individuals, as you say, developers, the community at large, but ultimately it can entrench a particular solution and it allows products to be built around it. Countless big businesses like Databricks have been built around this idea of having something open source available and then productized versions, maybe enterprise tiers with extra security, these kinds of things built on. And it makes so much sense to me. In fact, when this episode airs, I will have just returned from the open data science conference in San Francisco. And that's my favorite conference because you're mostly focused on open source solutions to AI problems as opposed to locking in with any particular vendor option. Isn't it kind of convenient? Or maybe it's just the way that you guys think at Cisco.

00:33:23 Cisco, when I think of that word in my head, it's in my embedding space in my brain, it's very closely related to just the idea of networking. I think that's what Cisco has been predominantly known for over the decades that Cisco's been around. And so it's interesting or convenient

or that you're focused right now at Outshift on these kinds of problems you've been discussing on the podcast so far. And we have very different kinds of problems we're going to get to next in the episode. But on the problems you've been discussing so far, it's been all about networking between agents. So that's kind of interesting, isn't it?

Vijoy Pandey: 00:34:03 I think of computing and scaling computing across three axes. You can scale computing across the axis of algorithmic improvements. You can scale computing using vertical scaling, which is building bigger and bigger of something, a particular something. But then you can also scale computing by horizontally scaling. So the three axes are algorithmic, vertical and horizontal. And that's how computing has scaled over the past seven or eight decades or so. And we at Cisco, we feel that the horizontal scaling axis is the place where we play best. So yes, you can narrow it down and say it's networking, but it's a broader statement than that. We bring in the aspect of horizontal scale to a computing problem that others may not look at. So others might be building bigger and bigger brains, in this case, bigger and bigger models or bigger and bigger agents and smarter agents.

00:35:04 But we are looking at scaling out super intelligence rather than scaling up super intelligence. You could be building bigger and bigger computers and servers. And that's what companies have been doing for decades. We invented the internet and we invented distributed computing and we are a large player in the cloud computing space because we allow for distributed computing to take shape through the networks that we build. So our thesis has always been to build secure, observable networks that take computing and horizontally scale it. We do work with vendors who are doing the algorithmic scaling and who are doing the vertical scaling, but it's our job to then horizontally scale that computing paradigm. So yes, it might seem convenient, but that's where we shine and

that's what we think are the problems that we are best suited to solve.

Jon Krohn: 00:36:02 Makes sense as a focus for sure. With you using so many driving analogies as we've been talking about this topic, correcting a vehicle as it seems to drift into a lane. I love that analogy. It makes a lot of sense as we think about agents working together and drifting away from that problem. But at the time of us recording, and this might be outdated by the time that this episode airs in about a month's time, but at the time of recording, I just became aware this week about a really cool new benchmark called Driving Bench. I don't know if you've heard of this, but it's a benchmark where they take general cognition models. So Claude Fable 5.1, Grok 4.6, GPT-6 Astra, and they have it in real time try to do a simple driving course through a parking lot where it just real time video feed input, the general cognition LLM has access to steering control, gas, brake, and pretty interesting.

00:37:10 So the results show that... So Grok 4.6, for example, they give each algorithm three tries in one continuous chat and Grok 4.6, the best that it scored across all three tries was it got 11% of the course, which is not even making the first turn. Same thing for GPT 5.6 Soul, but Claude Fable 5.1, a new bigger class of algorithms, I guess it's like a vertical scaling example. ClaudeFable 5.1 on its third try gets halfway through the course. GPT-6 Astra on its first try got halfway through the course on its second try completed the whole thing. And it's a pretty wild... It's hard to kind of wrap my head around how that can even work, how a model that is trained... There is a vision component to these models that's critical obviously to being able to drive, but that they're able to learn how to do something like driving in just a few shots, presumably without much or any explicit training on that kind of task, it's pretty wild.

Vijoy Pandey: 00:38:23 Yeah. And so I haven't looked at the benchmark yet, but is this... I'm assuming it's on a 2D map and you're steering through a 2D map or is it... I mean, I don't think they're actually driving a car in an actual world.

Jon Krohn: 00:38:40 They're driving a car. They're actually driving. A Toyota Corolla in a parking lot. Oh,

Vijoy Pandey: 00:38:44 That's

Jon Krohn: 00:38:44 Pretty amazing.

Vijoy Pandey: 00:38:46 I mean, that to me is also the next frontier where we are sort of reaching the capability curve on a 2D space, but getting into world models and where that direction is headed towards, that is the next frontier. And even there, like we were talking about earlier, yes, these models will get really, really good at individually steering a car or a embodied AI physical robot, but then how do these swarms or societies of physical embodiments work together to complete a task is actually going to be really interesting because you cannot have a singular entity do everything. So you can't have a singular robot. I mean, at some point that physical AI needs to transfer or do a handoff to a different entity. And whether it's handing off a box or whether it's handing off an incomplete picture of a car or something or the other, there needs to be a handoff between many, many, many physical embodied agents, robotics.

00:40:02 And it's the same problem yet again, because if you see these cognition failures in software systems and they could be silent with loud consequences, I can guarantee you that these failures will not be silent and will be really loud and scary when it comes to physical embodiment in the real world.

Jon Krohn: 00:40:24 Yeah. You touched on exactly what's most surprising about this, which is without having seen the research and

I'm not sharing my screen with you or anything so you can't see, you don't have the benefit of seeing the Toyota driving around, but it makes perfect sense to think that this would be some kind of simulation, some simplification, because that's what's surprising about this to me is that without this kind of explicit 3D world model training built in, this latest class of LLMs, Fable 5.1, GPT-6 Astra, were able to... The world models are inferred and can be represented within the embedding layers of these models that were trained on still images and text from the internet. It's pretty wild. And actually brand new news at the time of us recording this episode, the time of recording this episode is that Fei Fei Li's World Labs, business named World Labs has just been acquired by AMD for $8 billion.

00:41:29 And obviously, as you're saying, there's so much more to still be done in the physical world, but it is interesting that the scale of these models that weren't even designed to be able to navigate the world can figure it out a little bit now in some cases.

Vijoy Pandey: 00:41:48 Right. And I think the other interesting problem, again, to extrapolate is how can you then take these world models, take these text models or video models, and then again, align them, make them coordinate, go after a shared intent, and then solve for business outcomes that you and I do every day long. And what does that look like in a year or two years from now? And are we building the right infrastructure? Because one of the coolest things that makes agents, whether they're software or embodied, different from humans is that humans don't work on deterministic infrastructure, but agents still work on deterministic infrastructure. And so there is codification of rules and regulations and what's possible and what's not possible in software that you can do to then make these agents collaborate safely, coordinate, negotiate, and then work towards a common goal, which humans is not that easy to do.

00:42:59 So the fact that they are software is still a benefit. It's a huge benefit because we are not software and it's very hard to make us align our incentives. It might be easier to align agents on incentives.

Jon Krohn: 00:43:13 We sure hope so. And it's interesting you touching there around probabilistic and deterministic. It's a really nice segue into the exciting topic that I want to dig into for the remainder of this episode, which is around the computing substrate itself and how computing today is largely... I mean, what percentage of compute is done with non-classical systems today? It's negligibly small, but you do have some interesting examples about things like quantum computing that we'll talk about in a moment. But if we think about classical computing, which is today 99.999% of the compute that's happening on planet, it is deterministic, which is interesting because we are increasingly in a probabilistic computing world from a software perspective. So when you and I were talking about topics to discuss today, that is... You saying that to me yesterday as we were preparing for recording today, it's such an obvious thing.

00:44:22 It's such an obvious mismatch that had never occurred to me. So fill us in more on this idea of Outshift, the group that you run at Cisco that's so forward-looking, you're trying to prepare us, as you've mentioned earlier in this episode, for what's coming years from now. And it seems like one of the big opportunities coming years from now is to have probabilistic hardware to match probabilistic software.

Vijoy Pandey: 00:44:47 That's right. Right now, computing is going through a paradigm shift from applications all the way to infrastructure. And that paradigm shift is the movement away from deterministic systems to probabilistic engines. That shift actually happened in the workload at the application layer first, and we are seeing that all day long. So everything that you're looking at in terms of agentic

computing is actually a probabilistic engine. Your inputs are non-deterministic in nature. Give me the best or optimal answer. And what comes back is an optimal or a best answer with a certain confidence attached to it. So there's no binary here. So the workload moved towards a probabilistic engine, but the substrate that is doing the actual computation is still binary. It's still very deterministic. It's still matrix multiplications or GPUs. So there's a little bit of a mismatch between what's happening at the workload layer and what's happening at the substrate layer.

00:45:55 So it's almost as if the workload layer has gone to a dimmer switch where you can dim a switch between complete darkness to complete brightness, whereas a compute layer is still an on-off switch, which is either it's completely bright or it's completely dark. There's no something in between and there's no distribution in between. So we have to fix that gap because if we don't fix that gap, we will always live in this world of approximations and approximations cost, energy, money, size. And so that's where we are. So there's a whole new set of computing paradigms that are emerging to build computing as a non-deterministic engine or a probabilistic engine. And one manifestation of that is quantum computing. There are many other manifestations of how you can build non-deterministic compute.

Jon Krohn: 00:46:55 Love it. So let's talk about some of these specific implementations. So the one that I've already mentioned is quantum. I think that's something that when we think about distributions that we can be working over when we think about having that, and I love that analogy, it's so brilliant to think of it as a dimmer switch where we have this gradient from completely dark to completely bright. Yeah. So quantum computing is one of the... Why don't we talk about that first? And then there's other types of

probabilistic computing that I'd never even heard of before you telling me about it yesterday.

Vijoy Pandey: 00:47:27 So if you think about probabilistic engines at the compute layer, the idea for this can be credited back to Richard Feinman in 1981 where... And he's a famous physicist. He's one of my favorite physicists because he always ran with good analogies and good statements.

Jon Krohn: 00:47:48 I think he's a scientist that every scientist wants to be. I think he's just the best.

Vijoy Pandey: 00:47:56 He just aligns with what you would want to hear and he just grabs you and brings you in. So one of the things that when he was looking at these really large problems, and what do I mean by large problems? These are problems that really matter to humanity. So we see a lot of these drug discovery, new materials discovery, weather forecasting, financial modeling, logistics planning. These are really large problems. And what large problems have in common is that they run on really large spaces, state spaces, and the entire act of finding the best drug or the best material is actually a search algorithm that needs to run on this pretty massive state space. So that is what we are doing all day long. And so all the problems that matter to humanity can be boiled down to a search problem across a really large state space.

00:49:01 Now, to do these searches, you can do the classical route that we just talked about, but what Feinman said was a lot of these problems are nature problems and nature isn't classical. Nature is quantum. So why are we trying to put a round peg in a square hole? Let's use quantum principles and then build quantum compute to then solve for these nature class problems. And so that was the genesis of the entire quantum computing thesis, and we've been building that out. So we've been building computers that deal with qubits or quantum bits versus bits that we are familiar with. So a bit can be a zero or a

		one at any given point in time. A qubit is a zero and one at any given point in time. So the Schrodinger cat is both dead and alive at the same time because it's inside the box and you don't know whether it's dead or alive.
	00:50:05	So it is both dead and alive till you open the box. Whereas a classical computer is a dead cat or a live cat. We are hoping it's on a live cat, but it's a dead cat or a life
Jon Krohn:	00:50:18	Cat. It's just an analogy, people. It's just
Vijoy Pandey:	00:50:20	An analogy. Don't get too attached to
	00:50:22	It. So qubits are useful because since they can be zero and one, they allow you to explore this vast state space search very, very efficiently and quickly. So they can be really effective in finding new drugs, new materials, do weather forecasting, all of the problems that we talked about. The problem that we face today is we are still a little early in that journey. So for all of these big problems, not just to meet the bar, but to be actually pragmatic and practical at solving these problems, we need about a hundred thousand to a million qubits, physical qubits. What we have available today, the best case, is about a thousand physical qubits. So there's a massive gap between the qubits we have and the qubits we need.
Jon Krohn:	00:51:23	Yeah, two or three or four orders of magnitude.
Vijoy Pandey:	00:51:25	Three to four orders of magnitude. But as these quantum compute companies like the Googles and the IBMs and the Hatton Computing, Inflection, Qera, there's a whole bunch of these, as they build bigger and bigger quantum nodes, that's vertical scaling, as the software people in quantum algorithms do algorithmic scaling, we, again at Cisco, step in and say, "That is a path that is not being explored and that is horizontal scaling." So can we connect a hundred of these thousand qubit computers

and create a distributed quantum environment of 100,000 qubits? So can we build a quantum network that operates in the quantum realm to then be able to create a holistic logical quantum computer of 100,000 qubits? And that is what we are trying to do at Cisco. We are building a quantum network that can shift or switch quantum information between any pair of nodes without actually looking at or measuring that quantum information because that defeats the purpose.

Jon Krohn: 00:52:43 So it's the horizontal scaling from Cisco again.

Vijoy Pandey: 00:52:46 That's right. And one of the things I would say is the entire... Again, this is where networks are dramatically different, because networks for the past 40 years have been built on one principle, which is store and forward, which is I store a packet that comes in. I look at a packet header, decide where that packet needs to go, and then I forward that packet to the next destination. That's how the entirety of the internet has been built. Guess what? In quantum networking, you cannot look at anything. So there is no header to look at. There's no way to store the packet, look at it, and then decide where to send it. So that decision of where to send that qubit of information has to be done without measuring or looking at that qubit of information. And that's the challenge, and that's what we are solving for in Outshift.

Jon Krohn: 00:53:43 Really cool. And just like earlier in the conversation when I was saying how much of this is just a white paper thing when we're talking about these agent protocols versus something that my listeners can actually use, well, in this example, I guess my listeners aren't going to be able to be doing quantum networking from home immediately after listening to this episode, but at least there is something really happening here. So I understand that this idea of being able to have communication between quantum nodes is really happening. There's a Brooklyn Manhattan deployment, so I guess I actually live on the side of

Manhattan, just looking out over Brooklyn, and so it could be happening near me.

Vijoy Pandey: 00:54:26 Think of it as a separate kind of Manhattan experiment, Jon. It is a safer and better kind, but the way we've... So at Cisco, we've been building the entire quantum networking stack from the ground up, from chips to protocols, to controllers, to applications. And we've been releasing the chips separately, we've been releasing the protocols separately, and we've been releasing the application separately. And we've been doing that over the past couple of years. We've released an entanglement source. We've released a quantum switch just a couple of months ago. We've released a whole bunch of protocols, and these protocols might sound science fictiony, but they're actually true. They are protocols such as entanglement distribution. They're protocols such as teleportation detection. They sound like Star Trek, but they're actually true. And you can use all of these to build a quantum network today. Now, quantum networks have been built in lab environments for a long, long, long time.

00:55:30 What we decided to do with some of our partners, Qnect is one of the partners, QTD system that supplies fiber in New York City is one of the other partners, and then New York University is the third partner. So along with all of these partners, we decided to do a real world quantum network experiment. And we did that, I think it was February of this year, if I'm not mistaken. This experiment was done over 17.6 kilometers, roughly 11 miles, from Manhattan to Brooklyn and back. And it was done on real New York City fiber, the dirty, humid, hot, shaky, going in subway stations below buildings, below streets, that kind.

Jon Krohn: 00:56:22 Hey, don't talk about my fiber like that here, huh?

Vijoy Pandey: 00:56:25 It's real and it's true and it's amazing. And so think of these fibers as real world fibers. They are real world

fibers. And what we did is we said, "Let's run our network across 11 miles through 60 Hudson." This is a telecom carrier hotel, for those of you who are not familiar with carrier hotels. This is where a lot of the fiber from across the Atlantic and from across the US all converge. So 60 Hudson is one of the busiest carrier hotels in the world, and this fiber actually ran through 60 Hudson. And what we showed is entanglement swapping, which is the basic bitwise operation, if you can think of it that way, in the quantum world. It's entanglement swapping across these two ends, and we showed rates that are 5,000 times better, 5,000 times better than what has ever been done even in a lab environment.

00:57:32 So this network is real. It's in a New York City environment through existing fiber performing 5,000 times better than what you've ever seen in a lab environment. And the reason it could do that is because we are building this coherent stack all the way from chips to controllers and protocols, the software and the application all working in concert, not just with Cisco chips and hardware, but also with our partner hardware like QNX hardware and working on existing fiber like QDD systems. So we are building the quantum internet one link at a time, and this was the proof point in real life of at least 11 miles of a possible quantum internet.

Jon Krohn: 00:58:22 So this is going to be a really dumb question from a quantum newbie, but I thought the whole idea of entanglement is that we could be any distance apart and that entanglement is still in place. Yet in this situation, you're describing a physical, dirty, humid New York network cable that is somehow supporting entanglement?

Vijoy Pandey: 00:58:44 That's a great question because the way the entire quantum networking paradigm works is... The way I would describe it is let's say Jon, you're sitting in Manhattan, I'm sitting in San Jose. At some point, we need to exchange quantum state. So this is not actual

classical information because that is going to break a bunch of physics laws. So we are not really sending classical information faster than the speed of light. There is still the no cloning theorem and it still holds. We have not crossed that boundary because that would be tough. But what we can do is we can take coordinated action and exchange quantum state between the two of us. And the way that works is there needs to be a common source, almost like an Amazon. Let's say there's an Amazon that sends a physical box to you and to me. This is the generation of an entangled pair of photons.

00:59:50 One of the pair goes to you, one of the pair comes to me. So that is still very physical. So it is sent over these existing fiber links and it follows the speed of light principles. So that needs to happen. But once you get the box and I get the box, one photon each in this case, then the entanglement property can be teleported at distance. So now the property, the quantum information that sits in that entangled photon is actually an instantaneous piece of state information that is coordinated across the two of us. So that is a teleportation part, but physically you need to send these entangled photons and then after that, the state information is a teleportation problem. So that's the way to think about this. So as long as you and I stay in the quantum domain, think of it as like that movie Ant Man, I think is that movie?

Jon Krohn: 01:00:52 That's like a Marvel comedy, right?

Vijoy Pandey: 01:00:54 Yeah. So we've been shrunk to the quantum state and when we are in that quantum realm, at that realm, we can exchange information using teleportation. But if we want to come back to reality, to the real world, to the classical world, then we need to come back and we will not be exchanging information. We will be bound by the speed of light. So that's the way to think about it. But as long as we are within the quantum realm, the quantum

network treats all of these nodes and makes them behave as one singular entity.

Jon Krohn: 01:01:30 I can't candidly wrap my classical mind around teleportation or quantum. None of

Vijoy Pandey: 01:01:39 Us can. In fact, even Einstein couldn't and he called it spooky action at a distance. So you and I are just mere models and it's fine.

Jon Krohn: 01:01:49 Not as smart as Einstein or smarter than Einstein.

Vijoy Pandey: 01:01:52 And mathematically things work and in the lab they work. So we take it, we prove it and we run with it.

Jon Krohn: 01:01:58 All right. Well, I guess I can entrust an LLM to do it for me. Perfect. So that's quantum computing, which is a non-deterministic kind of computing substrate that I'm sure every listener has heard of, but tell us about some more niche ones that are also coming online.

Vijoy Pandey: 01:02:18 When Richard Feinman made the comment that we need to leverage quantum because nature is not classical, nature is quantum. So let's leverage quantum computing to solve for these really large state space search problems. He took a leap, which is almost the right hand side of that spectrum. So it's the right bookend. Let's leverage quantum physics to then solve for these state space search problems. We all know that there are other kinds of physics. There's classical physics. And so there is this spectrum in between, between classical deterministic binary systems and quantum systems that are classical physics based that can also carry distributions instead of carrying binary data and can solve for these search problems in vast state spaces. And so we are seeing a whole landscape of startups emerging to solve this problem. And they're leveraging classical physics laws to build silicon that does computation on distributions rather than zeros and ones.

01:03:39 So if you think about the computing spectrum, you have deterministic systems working on zeros and ones. You have quantum systems working on superposition of zero and one, and then you have the spectrum of computing in between that works on distributions between zero and one. And so some examples of that, to just wrap our heads around it, are thermodynamic based systems. There's this company called Xtropic is building these silicon systems, is actually shipping them now that work on thermodynamic principles to then solve for an outcome. And the way they work is they find the low energy state in a thermodynamic system. And when that system finds that low energy state, that is the outcome that you're looking for. So imagine you're dropping a marble in a valley punctuated by hills. And so when you drop a marble, it settles into one of these valleys and wherever it settles is the outcome that you're looking for.

01:04:52 It may not be the global outcome, but it could be a local optimum. So that was thermodynamic based computing, but there are other types of physical laws based computing. One pretty well known startup that has emerged in the past six months is unconventional AI. They've raised a lot of money to solve for silicon that works on oscillator based principles. So unlike thermodynamic based principles where the result is settling in on an energy minimum, here, these are oscillation based computer elements where once they align to a particular oscillation pattern, that is your outcome. So think of this as metronomes all swinging and once they all align on a particular pattern of swinging, that's where your answer lies. So again, these are not binary systems. They will all give you local optima, but that's exactly what you're looking for in these really large state spaces because you're looking for a local optimum and that's where your current answer is.

01:06:04 If you had more time, you could look for better local optimum, but you may never get to a global optimum

because they may never be a global optimum. There is never the best drug. It's just the best drug for the time being. It will always improve as our technologies improve and as our understanding improves. So you're always looking for the best outcome given the constraints of time resources that you have at your disposal. And that's what these systems are doing over a distribution, whether it's thermodynamics or oscillation based or over superposition in quantum sense that are different from the binary deterministic world that we are so familiar with.

Jon Krohn: 01:06:47 Really interesting. So yeah, so thermodynamics based, oscillation based, quantum based, three different emerging possibilities for non-deterministic computing to match the non-deterministic software that has become so prevalent with AI and large language models. Something to keep an eye on for sure and a really fascinating conversation. For our listeners, Vijoy, who want to be able to get more of your thoughts, I understand you have a reading recommendation for us that might help us out.

Vijoy Pandey: 01:07:23 First and foremost, the pieces that inspired us to look at this whole arc of human intelligence and then map it to the parallel that's happening in agentic evolution and towards super intelligence. There's work by Michael Thomasello. It's a pretty heavy read, but you have LLMs at your disposal that can summarize, make it easier to digest. Go ahead and do that because the work that Michael has done on the evolution of human intelligence and what we can learn from that is just fascinating. And you will see those parallels play out today every day. And it's not just me and Auschif that got inspired by it, but many, many academics, many, many frontier lab people, many labs in these large vendor environments, they're all getting inspired by that work. So it's something to keep a close eye on. I myself have pivoted from reading books to reading research papers, which sounds pretty boring, but that's where the frontier is moving.

	01:08:33	So I read a lot of archive papers and I keep up with those papers out there. But I've been writing a lot about where the world is headed on my own Substack, which is called Coherent Cognition. And the world actually talks about the cognition piece, the coherent piece, which is the quantum side and everything in between. So you can take a look at Coherent Cognition in Substack on LinkedIn, but also I'm starting a podcast on my own. And the thesis behind that is to make all of this frontier research that is not available as easily as possible to you and I, to lay people who are trying to understand and make sense of it all from the business perspective, simplifying that for business people, simplifying that for infrastructure people, for software people. I'm trying to do that by talking to researchers who are at the forefront of deep technology research.
	01:09:32	So all of these are valid ways to get familiar with what's happening and what's coming.
Jon Krohn:	01:09:37	Brilliant. Well, thank you for sharing your deep thoughts with us on the show for a third time today, Vijoy, and for giving us with resources to continue to follow along with after this episode recording. As always, really enjoy the time with you and hope to have another session with you soon.
Vijoy Pandey:	01:09:56	Thank you, Jon. It was just an amazing conversation to have.
Jon Krohn:	01:10:00	Well, that was a tremendous episode always love when Vijoy Pandey comes on the show. In today's episode, he detailed how agentic AI has moved from isolated agents to connected agents, but connection isn't the same as cognition. Agents can exchange valid messages without agreeing on what those messages mean. He talked about how every multi-agent session his team analyzed had at least one cognitive failure and that these failures compound so fast, the three of them drop accuracy to just

16%, all while your observability dashboards stay green. He talked about how Outshift's open source observability and explainability platform OXP works like lane assist in your car watching open telemetry traces for drift and triggering the Accord protocol to lock in shared intent or the Concord protocol to resolve conflicts mid task. He talked about how Cisco is scaling quantum computing horizontally by networking today's thousand cubit machines together, recently demonstrating entanglement swapping over 17.6 kilometers of Manhattan to Brooklyn fiber at rates 5,000 times better than anything achieved in a lab.

01:11:04 And he talked about why running probabilistic AI software on deterministic hardware is a mismatch from what we're doing today and how startups like Xtropic and unconventional AI are building chips that compute on distributions using thermodynamics and synchronized oscillators. Very cool stuff. I hope Vijoy will be back on the show again soon. As always, you can get all the show notes, including the transcript for this episode, the video recording, any materials mentioned on the show, the URLs for Vijoy's social media profiles as well as my own at superdatascience.com/1041.

01:11:39 That's today's episode number, yes, episode number 1041. Thanks to everyone on the Super Data Science podcast team for making all of these wonderful episodes. Our team today is podcast manager Sonja Brajovic, media editor, Mario Pombo, our partnerships team Natalie Ziajski, our researcher, Serg Masís and our founder Kirill Eremenko. Thanks to all of them for producing another outstanding episode for us today, for enabling that super team to create this free podcast for you. We are deeply grateful to our sponsors. You can support the show by clicking on our sponsor's links, which are in the show notes. It really does make a difference. And if you'd ever like to sponsor an episode yourself, you can get the details on how by making your way to

Jonchrone.com/podcast. Otherwise, help us out by sharing this episode with somebody who would enjoy the mind-boggling insights within it. Review the show on your favorite podcasting app or on YouTube, subscribe, but most importantly, just keep on tuning in.

01:12:37 I'm so grateful to have you listening and I hope I can continue to make episodes you love for years and years to come. Until next time, keep on rocking it out there and I'm looking forward to enjoying another round of the SuperData Science podcast with you very soon.