SDS PODCAST EPISODE 939: MIXTURE-OF-EXPERTS AND STATE-SPACE MODELS ON EDGE DEVICES, WITH TYLER COX AND SHIRISH GUPTA

Jon Krohn:	00:00:00	Mixture of experts, models and state space models are powerful new LLM architectures, but they are cumbersome to work with, particularly if you want to use them locally or on edge devices. Right? Well, not anymore. Welcome to the SuperDataScience Podcast. I'm your host Jon Krohn. Today I'm joined by Tyler Cox and Shirish Gupta deep experts from Dell who not only detail what mixture of experts and states-based models are, they reveal solutions that make using these state-of-the-art capabilities locally a piece of cake. Enjoy. This episode of SuperDataScience is made possible by Anthropic, AWS and Gurobi.
	00:00:37	Tyler and Shirish welcome and welcome back to the SuperDataScience Podcast respectively. It's great to have you on the show. Tyler, where are you calling in from today?
Tyler Cox:	00:00:49	Hey Jon, I'm calling in from Austin, Texas.
Jon Krohn:	00:00:52	Alright, and then Shirish, I assume you're also calling in from the Austin area, is that right? I am. Perfect. And the reason why I could assume that you're also in the Austin area is because one, you've of course been on the show before Shirish. In fact, episode number 921 of this podcast from September is one of the most watched videos that this podcast has ever had on YouTube over a hundred thousand views, which is very cool to see because kind of just started making a push into the YouTube format and it's nice to see that that took off. Very fun, very informative episode. In today's episode, you're accompanied by a different colleague from Dell. We'll see if we can get the banter up to the same level with Tyler. So Tyler, your title is Distinguished member of technical staff at Dell. What does that mean? It sounds distinguished.

Tyler Cox:	00:01:45	Yeah, so I actually got a recent promotion. I'm a distinguished engineer at Dell, but I work on all things on device AI for our client solutions group. Lots of IPCs all the way from our mainstream IPCs all the way up to our great workstation products.
Jon Krohn:	00:02:06	Very nice. Congrats on the promotion. And so we are going to be going into a really deep technical episode because you're here, I mean Shirish obviously can get big into technical details as well, but Shirish, I've got some opening questions for you kind of around framing the particular products and services that we're going to be talking about in today's episode. And then I've got some deep technical questions for Tyler to use his new distinguished engineer title with. So Shirish, when you were last on the show, we talked a fair bit about something called the Pro AI Studio, and I understand that that was created on a whiteboard with you and Tyler, but there's been some reframing internally around that depro AI studio and so it sounds like it's something that's in development hot off the press. Do you want to tell us about this?
Shirish Gupta:	00:03:08	Absolutely, and it's a nod to the whole team at Dell for something like something as complex as Depro AI Studio doesn't just come into being unless there's a whole village behind it, right. Special mention to a couple of folks by the name of Spencer Bull and Jacob Mink who worked on a very key part of Pro Air Studio before Tyler and I got to the whiteboard, so just want to call that out, those two special gentlemen out and their leadership. Alright, so to get to your question, you are absolutely right John, and if you recall when we spoke last Del Pro Studio is really, it's a toolkit for our customers. It's not a product unto itself and which is one of the main reasons why we are rebranding it. We're in the process of rebranding it and so don't know what it's going to be called, but just know that in short duration from here it will have a different name.

And why that's important is in the construct of the Dell AI factory, the toolkit formerly known as Dell Pro AI Studio. It really extends the capabilities of the AI factory to our entire edge and PC portfolio. And that was a critical part of the reason why we actually brought it into being

Jon Krohn: 00:04:49 Nice. And so this thing that was formally known as the Dell Pro AI Studio in the previous episodes that you were in, that's what we called it, it's going to be part of now a bigger offering called the Dell AI Factory and what exactly it's called, we'll probably know by the time this episode comes out, and so I'll be sure to make that really obvious in the show notes on whatever platform you're watching or listening to this episode on. Keep an out there so that you can easily look it up, we'll have links to it and you'll be able to understand everything about it regardless of whatever it's called. Tell us about the origin story and what problem you were trying to solve.

Shirish Gupta: 00:05:31 Yep. I mean this journey started, I'd say arguably early last year and everyone in the business was on an AI journey. The new buzz was still developing very rapidly, so not everyone really knew what they were doing, but everyone knew that it was a very urgent journey to be on and we wanted to hear from our customers and hear what problems they're trying to solve and how they intend to use ai, especially on our edge and PC devices. So a lot of research, a lot of customer conversations led to the clear realization that we needed to make it real for them. What was on device ai? That was the big question when we started out a year ago, and since then, and we've covered this briefly in one of the earlier podcast episodes, it's ultimately comes to the PC fleet in the form of software that's managed on an enterprise, any enterprise catalog software catalog, which customers can either buy software directly from independent software vendors like Microsoft, Adobe, CrowdStrike, McAfee Next Think Citra, et cetera, or they can build AI solutions that are

embedded into their workflows and solve problems that use their own data in a very private and proprietary or sovereign way.

00:07:17 So the short form of this long-winded context is the software, the a i PC solution that is formerly known as depro Studio was formed in order to help customers solve problems that was making it extremely difficult for them to bring their own custom AI workloads and run them on device on their PC fleet.

00:07:47 And some of the problems that we learned that we needed to solve was one, this inherent complexity with bringing models and running them on PCs, diversity of silicon, diversity of runtimes and execution providers. There wasn't at that time a really good solution that would abstract away all of that engineering backend complexity for developers. And we started seeing that it was slowing down adoption not only for customers but also for ISVs. So we had to solve for that. The second thing we saw was there were a lot of tools that was starting to crop up that did allow you to bring AI and run models and run them on device, but there wasn't anything that truly was enterprise grade. And what I mean by that, just as an example, is it's not enough to just bring a model and run it locally on one pc. You need to have the ability to deploy and manage that through the entire lifecycle off that AI workload or app across a whole fleet of PCs and it needs to come with the manageability and security frameworks that enterprises are familiar and expect.

00:09:08 So that was the second problem that just didn't exist for on-device ai. So we knew we had to solve that. And the third thing we had to make sure we did was that we had to make it as easy as deploying AI in the cloud because everyone was racing to value creation and that means what was the easiest path of realizing AI value and outcomes and a lot of those workloads started in the

cloud and customers still start their journey there. For the most part, if we had to have a chance of bringing workloads reliably on client, we had to make it as easy as deploying on the cloud.

Jon Krohn: 00:09:50 So lots of our listeners will be familiar with how easy it is to call an API and use an LLM from a third party proprietary provider like a cohere or an open AI or an anthropic. Very, very simple to write an API call and have this model magically you provide some context, you provide some instructions and magically you have results. One step more complicated than that typically historically has been being able to deploy your own model to the cloud. So regardless of which cloud provider you use, there's lots of functionality there that makes it easy, relatively easy, not quite as easy as calling in API, but pretty easy to have your own model running up in the cloud in kind of like a serverless kind of feeling. Or you could have it running on a server depending on your particular workload constraints, but historically having the models run on the edge or on local devices like PCs, which are the most common kind of computer in the world and which could benefit from it being easy to run models locally on those devices, it sounds like the solution that you came up with, this A IPC solution now allows that to happen, so it makes it as easy as deploying to cloud to deploy locally, getting around all the kinds of concerns you mentioned like security as well as just all of the different kinds of devices that people can be using, cpu, GPUs, neural processing units that we've talked about in previous episodes with you Shirish on the show.

00:11:33 Did I kind of summarize that back to you accurately?

Shirish Gupta: 00:11:36 That was pretty good. Right, and so let me tell you a little bit about the components of what used to be pro AI studio, this AI PC solution, it's really three parts. One, it's a model catalog and it's hosted on hugging face inside

Dell Enterprise hub, which is where we also have models for enterprise, just like you said, serverless. So the same exact place where you can get models, larger models for on-prem deployments, you can now find models for the PC that service a variety of use cases, right? Text image, voice. The second key part, which is where those esteemed gentlemen I referred to earlier come into the equation is everything that happens on device, there is a AI framework that provides APIs and that actually ends up making it super easy for existing cloud workloads to just be pointed to the PC as a local host using existing API standards.

00:12:55 So that is the, again, I think goes back to my point of making it really easy to deploy workloads, and then there is also all of the model management that happens on the device. That's another model management service that is where the secret sauce was being built already within the organization when Tyler and I kind of brought it all together. And then the third part again resides in the cloud, which is our partner portal, which is accessible through Intune. It's called Dell Management Portal, and this is where enterprises have that lifecycle manageability capability for all of their enterprise AI deployments that were integrated with the Dell AI PC solution, formerly known as Dell Pro S studio. Right? I think the other cool thing that we've done, which no one else has picked up on yet, is that we've made every component modular and behave very much like any other app that enterprises would deploy through their existing app management consults. So both the models, the frameworks and the installers, the core service, they're all bundled into a single installer that they can just push scripted or click to publish through Dell management portal, which is I think really what enterprises need it teams need.

Jon Krohn: 00:14:31 Fantastic. So this sounds like a massive initiative and as you mentioned, there are lots of people involved that have

allowed this initiative to happen. Quickly. Before I get to you Tyler, and start asking you about this architecture and how it all works, either of you, I suppose just really quickly to remind our listeners, why should people be considering running workloads locally now? So we talked about this a lot in episode number 921 with Shirish and Ish. So if people kind of want to go over all of the detail, kind of like an hour of why you should be considering deploying workloads locally, we will do that. But just really quickly in a minute, if one of you wouldn't mind providing an overview of why we'd want to do that?

Shirish Gupta: 00:15:21 Well, that's a great question, Jon. I think if you think about if from the lens of a customer, if they're on their AI journey already, which most customers are today, unless they've explicitly made the decision not to embark on that gen AI path, but those customers that are on the path firmly, what on-device AI does is it gives them optionality, right? It's become very clear to me that most workloads or most experiments are starting in the cloud. Customers are doing POCs, they don't want constraints when they're doing POCs and when they start deploying to production, they want to get the value and outcomes and not be constrained by capabilities of models or compute, et cetera. So those customers that have actually gone through and deployed use cases and started scaling them out to their user base or doing multiple use cases, what they're seeing and they're signaling to us is that they're starting to realize that not every inference is equal and they're going to run out of either, depending on their model, their business model or how they've implemented ai, they're either going to have costs that are going to skyrocket as they scale out through their organization or if they're doing it on-prem, they're going to run out of compute space.

00:17:01 We know that the bulk of workloads are going to be inferencing in the next few years. Almost the prediction is

90% of all of AI workloads will be inferencing by the turn of this decade. So we believe, I firmly believe that the future of AI is hybrid and workloads will run seamlessly across the cloud, the edge and the PC fleet. So going back to my point about all inferences not being equal, that's a great point for AI leaders to start thinking about what is the right compute engine for the inference that they have at any given time. It to the concept of quality of tokens and what that really means is the quality of the model needed to get a quality response for a particular task can be vastly different. And if I take one scenario of code generation or code assistance, that's a use case that is really prevalent out there and we hear about it all the time.

00:18:15 So take that use case for that scenario. If you're doing auto complete, you can probably do it on a smaller model, then why burden a GB 300 super chipp with that? Why not do that on the pc? And then yeah, if you're doing a full scale code refactor, then sure you may need a frontier model, right? So what I think will happen is there will be a lot of intelligence that goes into deciding where workloads will run, but just that if you extend that thinking alone and play it out, I see PCs playing a pretty important role in the future of enterprise AI strategy.

Jon Krohn: 00:19:00 Nice. Okay. And then so as kind of concrete examples, just kind of reeling off off the top of my head, there'd be situations where you gave examples there where you just don't need a gigantic model running, so why not run it locally? You could save money, you could save time on bandwidth. There also could be lots of situations where you're on a factory floor and there isn't any internet connectivity or there isn't high quality internet connectivity or where performance in real time just matters so much you don't want any latency. So yeah, I dunno, just kind of reeling through some reasons why you might want to...

Shirish Gupta: 00:19:41 Yeah, and beyond cost, just like you said, if you recall I introduced the A IPC mnemonic back in episode 877, right? That's still one of my favorite contributions. That's my mic drop moment. Just walk away ish. Your contribution. But jokes aside, right, it's still a great way of recalling the benefits of AI inferencing. So like you said, there's plenty of use cases where latency is the most important, especially when it involves voice or video for human consumption. Humans can pick up those lags very quickly and it creates a poor experience. So doing some of that compute locally makes a ton of sense on the PC as you take away that 200 millisecond cloud latency. The other thing, of course, privacy of data. This is probably understated and poorly understood today, but to me, even if your data is not completely the device, very few of us have data only on the device.

00:20:50 A lot of our repositories are inside our firewalls, but they're in the cloud. Even in that scenario, if your LLM or your model, it's processing locally, your query remains completely private. And this is a big deal because if you're using a cloud model public cloud model, your query is completely indexable by search engines and people can know everything that was asked and everything that was put into the context input, output, context, everything. So think of any sensitive or data that inherently needs to remain private contractual obligations, contracts, IP or private sensitive queries to an HR database. There's there's a few examples. Even though there is a database retrieval request going through the network, the input and the output from the model stays completely private to the user on the box. So that's a pretty big deal in my book. And there's a new one that's not in the IPC mnemonic, but it made this recent AWS issue made me realize that this AI on your box is always available. You're not reliant on the wireless networks or browsers for cloud.

Jon Krohn: 00:22:19 Right, right.

Shirish Gupta: 00:22:19 Wherever you are, whatever's going on with the network, you always have access to your AI companion.

Jon Krohn: 00:22:24 At the time of recording, we've just gone through a big AWS outage where lots of services, for example in New York, I couldn't use the shared bike scheme here and I had to go around on foot everywhere for a couple days, which was weird for me of that AWS outage if only they'd been running and no, that wouldn't work in that scenario. Having local at each bike dock station, having a model running doesn't help because you need to know which bikes are available and which aren't, but yeah. Okay, so we have the context now around the value of doing local compute and both of you, Shirish Tyler, as well as lots of others at Dell, have come up with this brilliant solution, again formerly known as Dell Pro AI Studio, and now it's this AI PC solution that we'll have a name for you in the show notes when this episode comes out. And so it sounds like it's a huge initiative in order to have this work across all different kinds of hardware and have the security that people expect and be easy to do. People have gotten used to with cloud deployments or with just calling an API Tyler distinguished engineer, how is it all possible?

Tyler Cox: 00:23:38 Yeah, so we've got a great ecosystem of partners from the Silicon Model partnerships, application partnerships, and what we saw as we mapped out the journey is that we were in the right place to make a difference. So we're pulling together a lot of great technologies that cover a lot of the different pieces and trying to simplify it for our end customers and developers. So a lot of this is glue, right? It's making sure that when you're on a Pro Max system versus a Del Pro plus that we're solving for the different hardware ecosystems that are inside of that and presenting that consistent simple API across those so that

you don't have to know the different details of the quantization parameters and the tool chains that were used to prepare the models for those different platforms and the various runtime components that are in there. We've got that right now. We're at eight different backend runtimes and counting. Those are not things that we built. Those are things that we're integrating and solving for you. So your app doesn't have to carry the difference between an A-M-D-N-P-U and an Nvidia RTX Pro GPU and the different models that they need. So how is we don't operate on our own. We're bringing together a rich and robust and innovative ecosystem and trying to simplify on top of that.

Jon Krohn: 00:25:18 Nice. Well, that sounds great. I guess it takes a whole village to raise a little software child and so yeah, there's other kinds of features that sound complicated, multichip support model management, command line tools. How do those kinds of things, how do those kinds of features support people like our listeners who are hands-on practitioners, data scientists, AI engineers, software developers, how do features like that support them in their workflows?

Tyler Cox: 00:25:46 So just take 'em one at a time here. So multichip support, we have systems that have four or five different flavors of accelerators in them for an application to be able to route traffic appropriately across those to be able to say, Hey, I want this model running on the NPU for this session, but on the GPU for the next session, that's quite a bit to manage on a system that's going to cover a couple percentage of your deployment base, but for the end user being able to take advantage of all those different silicon types on the system they buy, that's really important to 'em. So what we've done is we've tried to make it simple for applications of all kinds to take advantage of the best local resources that are available and mask some of that complexity. Now, on top of that, we've built robust model

management framework so that we're making sure that the right versions of models and all of their dependencies are getting to the right platforms.

00:26:54 Cherise talked about our Dell management portal integration that's making it easy for the IT admin to deploy, right? So you're loading up bundles of our solution, you're attaching 'em to applications, and then when you go and deploy it to 5,000 users, you're not having to hunt and pick and pick and choose different versions of the application to land on those different platforms. We unroll that complexity for you. We make sure that the right versions of the models and all of their dependent packages are landing on those systems and showing up underneath a consistent application command line tools. That's really part of our developer and management experience so that you can get granular. That's something that we're not reinventing. We're making sure that we're leveraging some of the best practices that are out there so that those of you who are familiar with local AI development, you're going to see very similar patterns in using AI Studio or the formerly known solution in your development experiences. So we're trying to make it easy for you and also easy for enterprises to deploy.

Jon Krohn: 00:28:02 Excellent. Alright, so in addition to all those kinds of features, let's talk about particular models that might be interesting. So this is something that I think people like to hear a lot about on the show or in general is latest models, latest capabilities. And so Dell recently announced a partnership with IBM around their granite models, so granite like the stone, and so can one of you give me the big picture around what that partnership is all about?

Shirish Gupta: 00:28:32 Absolutely. We're pretty excited about that partnership with IBM, which we recently announced at IBM Tech

Exchange. And what I'm most excited about is through that collaboration, we're actually bringing really best of breed models into the AI factory that then customers can use to build, deploy, and scale rapidly across not only Dell's entire hardware ecosystem, not only from the infrastructure servers to edge devices to PCs. What I really like about these models, and we are talking specifically about IBM's granted 4.0 family of models, is that they're covered under the standard Apache 2.0 license. They are a hybrid architecture model combining state space models with transformers, and that really changes the game in terms of long context performance and memory efficiency. And these are enterprise grade and why do I say that? Because yes, one, they're open source under the Apache 2.0 standard license. They're trusted, they can be trusted open weights, open training data for full customizability and deployment flexibility and to top it off, they're performant, right? They're near the top of the leaderboard of the standard helm IFE valve for open weights and actually bookended by a hundred billion parameter plus models on both sides. So these models really punch well above their weight and in my opinion, their ideal starting point for fine tuning and customizability for enterprise deployments. Tyler, why don't you share some more about the models themselves?

Jon Krohn: 00:30:25 Yeah, let's talk about, that was a really helpful overview and it's great to hear that it is doing so well on the helm. I'll have a link to helm in the show notes for people that want to understand more about that. And so this is specifically at this time, this is the fourth version of this model family. So it's the Granite 4.0 model family and I

Tyler Cox: 00:30:42 Fourth, fourth major version,

Jon Krohn: 00:30:44 Fourth major version, right? Yes. Thank you for, yeah, it's good. We have software people on this call to get me right on my model. And so within the Granite four model

family, there is coincidentally four different variants, and so maybe you can fill us in on the details of those four variants and why you might choose one of these models for a particular use case. Tyler.

Tyler Cox: 00:31:09 Yeah, and there's a reason I interjected there and it was that what IBM's been doing with Granite is really exciting for enterprises. They're delivering some pretty consistent results and I'll start with the bottom of the family and highlight that, right? So Granite, the three series had a 3.0, a 3.1, a 3.2, a 3.3 version, all improving on the same model. The first model in the Granite for family extends and improves on that again, so that's the same familiar pure Transformers model architecture with new knowledge and improve performance. So it's a great 3 billion parameter dense transformer model. Then from there they get even more interesting. So the Granite four model family has the other three all have a H tag. So they are granite for H models that H there stands for hybrid and specifically it's a hybrid state space model. So what the IBM team has done is they've blended state space models, which I think we'll talk about here in a minute.

00:32:27 So I'm going to leave that definition hanging and transformers to get great long context scaling specifically one of the big things with attention mechanisms built into transformer models, one of the big limitations is memory scaling. There's a quadratic scaling law that means if you double the input context, you are quadrupling the memory you're required to run. For example, your KV cache in most scenarios, what a state space model, one of the important properties it has is you get linear scaling. So with the granite four H family of models, you have sub quadratic scaling there. So the first of the granite four Hs is a granite four H micro that accompanies the Granite four micro that I just talked about. So that is also a 3 billion parameter dense model, but now we have hybrid state space. Then the last two models in the family,

granite four H tiny and granite four H small are both hybrid state space models like the micro, but now they are also mixture of experts.

00:33:37 So these are sparse models. They have a total parameter count of 7 billion for the tiny model and 32 billion parameters for the small model. They have activated parameter counts that are lower than that. So it's a 1 billion. So on tiny you've got 1 billion active parameters on small, you've got 9 billion active parameters. What that means, for those of you who a sparse or mixture of expert designation is a new concept that means that they have a knowledge and accuracy profile closer to their total parameter count, but they have a throughput performance profile that is closer to their activated parameter count. So what they're doing is they're picking portions of the model dynamically for each request and really at a token level to answer and retain only the most important pieces of the model in answering a given request. So really interesting blend of technologies in that granite for family.

Jon Krohn: 00:34:43 The mixture of experts approach is certainly something that all listeners should be familiar with. If you've been listening to the show for years, you've already heard about it a number of times, but if you haven't, you definitely need to know it. In our space it's important because it provides exactly that kind of power that Tyler was talking about there, where you get the power, the capabilities of very large models or as you said, all of the available parameters in the model that you choose, but you get the speed and cost performance of a much smaller model because you're not using all of the parameters in the model on any given call, you're using a subset. It might be common to use an eighth of the parameters in a given mixture of experts model when you're actually running it inference time. So yeah, really cool approach. And so my understanding is that you

mentioned earlier on in this episode about the Dell Enterprise hub on hugging face, and so all of these granite models are available easily through that infrastructure, and so you can give me an answer to that while also telling me how Dell optimized these granite models to run on IPCs and workstations.

Tyler Cox: 00:35:59 Yeah, so Dell Enterprise Hub is our partnership with hugging face. If you go to Dell hugging face.co, you will see our model catalog there. That's what we're talking about. What we've done in Dell Enterprise hub is we've number one, all of the Dell technologies portfolio devices from AI servers through workstations, through PCs, they're all show up there. So if you say I have a Dell, Dell Pro 14 premium, what can I run? Dell Enterprise Hub is a great place to go. So inside of there, yeah, we have the granite for family of models. We had day zero support from AI servers to A I PCs on there. So great place to go to get started for optimization. So like I mentioned before, we go in and we run all the models on our devices, we figure out what the best set of inference time parameters are to fit different footprints of different devices, whether that's context length or some of the other things, and we'll bake that in. So we're presenting an API interface to applications. We're picking some of the best parameters for different devices so that you don't have to worry about whether you're on a 16 gigabyte A IPC or you've got access to something with an RTX Pro 6,000 Blackwell and 90 68 to vra. We'll make sure that you're getting best performance out of that model.

Jon Krohn: 00:37:39 Nice, that sounds like a useful feature. Now let's dig more into what you were talking about there. We dug a little bit now into mixture of experts. Now I want to dig into state space models and hybrid architectures, which you touched on Tyler. So the Granite 4.0 release introduced a hybrid architecture that combines state space models with transformers, and that sounds important too, but

maybe I've gone too far. Maybe we should start with digging a bit more into what state-based models are first.

Tyler Cox: 00:38:12 Yeah, so state-based models are a really important family of models, even pre AI usage. So if you go back to the 1960s, these are being used in space flight control. They've been used in population studies and economic modeling for a long time. The core construct is that you have an input signal that you map onto a hidden or latent state space with one set of equations and then you have a second set of equations that translate that state into an output that's observable, right? So there's lots of different ways that you can construct a state space model to represent different problems.

00:39:04 What's happened over the last 10 years or so is that state space models were investigated is part of the deep learning kind of revolution is how do you construct your matrices for state space models to better model different tasks without as much classical feature engineering approach to it. So that was one important thing, but then there's a couple of researchers at Carnegie Mellon and Princeton who really kind of drove this home over the last five years or so. So there's a great set of papers. I invite everybody listening who wants to know, learn more to look up the work here on mamba and structured state space models and things like that. There's a series of papers from 2021 all the way up to 2025 still working on it that introduced some really great optimizations inside of state space models to make them appropriate for sequence transformation and language modeling tasks.

00:40:19 So you get this evolution of structured state space models, your S four paper, and then you go into structured states-based models with selection and computation by scanning your S six, which turns into, hey, that's a lot of s's, sounds like a snake. Now we've got mamba and the mamba architecture really optimizes

state-based models for the kind of compute profile that's needed to be relevant for the language model tasks that we're applying here. And so some great optimization work that happened there on some great mathematical insights into the matrix properties of those, and I won't be able to do those full justice here, but really, really interesting work that kind of accumulated into the Mamba and mamba two language model blocks that IBM pulled in to the granite four H series models, right? So composition here you've got a nine to one ratio of mamba layers to attention layers in the granite four H family, so quite a bit of state space model in that hybrid.

00:41:43 Those are, like I said earlier, those are linear context scaling. They are the granite form H models are no positional embedding. So they have in the dataset out to 512 K context represented in the samples, they're validated out to 1 28 K. The IBM team says, theoretically you should be able to push it past that, right? Some really great long context performance. I think one of the key measurement points in the release notes are if you take eight sessions at 128 K context on a micro, so a 3 billion parameter model, you get about a 15 gigabyte of memory usage versus about 80 on a pure transformer architecture, right? So some really great context reduction, which means that on more constrained devices on the edge you can make use of more useful context in rag workflows and multiterm workflows and things of that nature. Just to close here, she had mentioned if f eval, that's a really great benchmark for instruction following structured output tasks. The other things that granite scores really well on is the Berkeley for function calling leaderboard, specifically BFCV three that it shows up in the top five as we sit here recording among a bunch of other frontier models and hundreds of billion parameter models even for the small 32 billion parameter footprint. So really, really punching above weight class there.

Jon Krohn: 00:43:38 Really cool. Thanks for all those stats in there. You really packed 'em in. And so it's kind of like a key takeaway basically by having these post transformer architectures as part of these granite models, it sounds like specifically they've gone with Mamba as their particular state space model. It allows us to have better performance over long context windows, like you're talking about half a million input tokens being able to handle that. And when we handle that kind of large amount of context, that large number of input tokens, so we start to become worried about memory and about compute performance. And so over these very long context windows with these state-based architectures like Mamba, we're able to have scaling that isn't the quadratic scaling that we were used to with transformers. And what I mean by quadratic really quickly for listeners is with the transformer architecture, which is still the predominant model architecture within LLMs today, as you increase the number of input tokens, the compute and the memory increases quadratic by a squared factor basically, and that starts to add up really quickly.

00:45:03 And so we're getting quadratic memory and compute performance, which matters a lot in these very large input token situations. So really cool. So you get a lot of power, a lot of flexibility with these Granite 4.0 models because they incorporate the state space models like mamba into it. And so I guess my final question for you on this, and it was actually kind of where I opened around this was I was talking about how the Granite 4.0 models are actually a hybrid that include, I think you mentioned there are a nine to one ratio of mamba versus traditional transformer attention heads. Why? What's the advantage of having both?

Tyler Cox: 00:45:45 Yeah, it's a great question. So one of the things we do see with Mamba is it has great global context performance, but it loses a little bit of the sensitivity of local context

that attention players are really known for right now. One of the other interesting things that happened right after IBM launched the granite four models is there was a great meta fair research paper that compared some of the performance of a pure transformers architecture, a pure mamba architecture, an intra layer hybrid and inter layer hybrid. And so the intra layer versus inter layer that's looking at running through attention layers and mamba layers in parallel versus interlay, which is what the granite four H models are, which is running through them sequentially. So I think one of the recommendations from that team was build more of these hybrid models, the training efficiency, which we didn't talk about, and the inference scaling efficiency is really great without a lot of trade-offs on accuracy given some of the mamba improvements. So I would expect to see more of these type of hybrid models in the future.

Jon Krohn: 00:47:10 No doubt, no doubt it is the future indeed some, yeah, a post transformer architecture will be the dominant architecture in the future. All right, so changing topics here a bit from all this really cool technical stuff to kind of the real world implications of this. If you are an enterprise and you're trying to take advantage of ai, there was a recent study, I talked about this on an episode in the past probably several times, but I did a whole episode, episode 9, 2 4 on this MIT researcher claim that 95% of enterprise AI projects fail where fail means that it does not deliver a return on investment in production. So either it never makes it to production or it's never profitable in production or it's never successful, it never meets your success criteria in production. And so can you speak to or provide examples of what typically goes wrong in enterprise AI projects and how our listeners perhaps leveraging things that we've already talked about in this episode can prevent those kinds of failures?

Shirish Gupta: 00:48:18 That's a great question, Jon, and honestly, you have done a stellar job covering it in that podcast episode that you referred to, but my take on it, I think some of the most important things that everyone that's on their AI journey will do well to sort of ground themselves to is it starts very in a very cliche manner, it does start with setting clear goals for what outcomes are desired, and that includes very specifically what is the desired type of response and accuracy of set responses, and I think we're still very, very reliant on humans for ensuring that outcome, right? In other words, you just can't proceed unless you have humans in the loop and expect to have at least during the training phase and expect to have good results. In fact, I think having humans in the loop even during production as feedback baked in to the solution itself is vitally important because you want to account for drift.

00:49:37 But another cliche I will share is don't apply gen AI for the sake of it, use the right tool for the job. I think I've seen plenty of misuse where everyone just wants to apply gen AI to a problem that is very well solved with traditional, I call it traditional, it's funny, but machine learning, solid, solid machine learning, deterministic outcomes. So those are the basics, right? I think beyond that, you just want to really be very deliberate about the problem you're trying to solve, right? So again, problem goal tool for the job sounds pretty basic. You could actually take that response and apply it to literally even a classroom discussion on how to do a project or how to manage projects. So I think those are the basics that you need to ensure, and then as you go from there, there are obviously going to be very use case specific task specific best practices, which need to be uncovered and adhered to as you go along the journey. One other thing I will mention here is that it goes back to some of the pain points that we anchored on earlier on in the conversation about the problems that we were solving for with the de

AI factory and the entity formerly known as depro a studio, and I'll defer to Tyler there to talk about what are some of the best practices that customers can take advantage of with the solutions that we are providing.

Tyler Cox: 00:51:29 So I think taking advantage of the device through our solutions means that you've always got a enterprise ready model at your disposal, right? There's 2 million models on hugging face. They all do different things really well. We're trying to bring the ones that we think are really great for a lot of key enterprise use cases ready to go. We also make sure that they're performant on our devices of all types, so when we pick models, we try to make sure that they've got great coverage across different silicon types, so they're really good building blocks. A lot of times you'll start down a pathway with one model and you might hit a dead end when you try to scale up or scale across different device categories. It can be a little bit challenging to figure out what the right pathway is to getting everything hooked up the right way.

Jon Krohn: 00:52:34 All right. That all sounds great Tyler. Shirish you mentioned to me before we started recording and you mentioned actually a little bit in today's episode as well, how Dell PCs perform on benchmarks, how Dell IPCs perform compared to previous generations, particularly around compute battery life efficiency. Do you want to fill us in a bit on those key benchmarks?

Shirish Gupta: 00:52:59 Yeah, absolutely. I think it's important for customers to understand what they're actually getting when they invest in the newer hardware, right? So for example, the Dell Pro plus with Intel, excuse me, core Ultra 200 V series chip, there's just tremendous benefits advantages over say an N minus two, non A IPC chip set, like the core ultra 14th gen. So I'll just give you some examples. 88% more battery runtime running Microsoft teams meetings compared to the non A IPC on the DEPRO plus with Intel

Lunar Lake 4.8 X higher graphics performance. So very important to think about the GPU as well. The IGPU, which is tremendously is shown tremendous gains, gen over gen and even within the gen of Intel's latest chip sets, lunar Lake with its system on chip architecture is just tremendous in terms of its gains for the IGPU. Not to be ignored, we've talked so much about NPUs, but I think if you look at the overall C-P-U-G-P-U-N-P-U as a whole, it's just a tremendous value proposition, almost 10 x on device AI CPU performance and 4.2 X AI GPU performance on that same device relative to the non A IPC.

00:54:46 And then if you really want to talk about specific models, that's what your users are going to be most interested in. The same device with Lunar Lake provides greater than six times higher performance on MRS seven B and LAMA 3.1 for text gen and after 5.6 times higher performance for Imogen with stable diffusion 1.5, just showing you the breadth of the AI performance and the battery and runtime and power consumption improvements for these devices relative to some of the non A I PCs. So I really encourage people to look at these new chip set architectures and the new devices from a holistic lens and not just focus on the NPU itself.

Jon Krohn: 00:55:39 Yeah, so it sounds like we've gotten to a point where companies like Intel with the way that they make their CPUs as well as obviously GPU providers, neuro processing unit providers, and then even companies like Dell, the way that they package all of that up with something like an A IPC, you are optimizing everything for AI workloads, for training AI models, for realtime inference on the edge on IPCs, and so that's how you're getting the big multiples that you just went through in dozens of examples of Shirish.

Shirish Gupta: 00:56:12 Yep.

Jon Krohn: 00:56:13 Nice. Alright, so final technical question for you. Historically, AI transformation was a multi-year, huge undertaking for enterprises, but that kind of timeline is irrelevant these days because if you think about a multi-year timeline, we have no idea what the capabilities are going to be like a few years from now. We can't be thinking about a solution today that's going to be ready in multiple years. How does the Dell AI factory accelerate the timeline so that we're talking about weeks or months instead of years getting to deploy AI models?

Tyler Cox: 00:56:49 Yeah, that's a great question. I think part of it is you've got to simplify how you think about it. If you can't get, you're down in the details on everything, then you're slowing down. So picking the right technology partners matters. It will help you go faster. The second thing is you need to build on technologies that will scale and help you not just at deployment time, but six months a year, two years down the road. We all know AI is not sitting still. The model you deploy today is not going to be the same one that you deploy in six months or a year. Having manageability built in means that you can go out and control that, right? If you want to go do an update to your application, give your user five or 10 or five, 10% performance or two x performance, we don't know what tomorrow brings, then you need to build on top of the right tool sets.

Jon Krohn: 00:57:53 Awesome. Really enjoyed this interview with both of you today Shirish and Tyler Shirish. We've had enough book recommendations from you already. Tyler, what have you got for us?

Tyler Cox: 00:58:04 Yeah, so I read a bunch of different things. I use it as kind of escape, whether that's fantasy, science fiction. One of the things I think is something that, a story that a lot of people will love is one that is being adapted, right? So Project Mary from Andy Weir is my recommendation.

		It's a great, great fun story, touches on a lot of interesting environments and excited to see the adaptation. I think next year is when that will come out to film as well, so make sure you read it before, before the
Jon Krohn:	00:58:49	Theater. Yeah, exactly. Alright, thanks for that recommendation, Tyler, and we might as well just stick with you for a second. How should people follow you after this episode to get more insights on, I mean, yeah, you went into huge technical detail on models on capabilities. Where can they get more insights from you after the show?
Tyler Cox:	00:59:09	Yeah, I, I'm on LinkedIn, we'll put the profile in there. I will say I'm not a heavy poster, so don't expect to see a ton from me, but you can be sure that you'll get the latest updates on the de ai factory from my feed.
Shirish Gupta:	00:59:25	Excellent. And Shirish same. I'm also on LinkedIn. My link will be in the show notes and that's maybe a little more active compared to Tyler, but still not one of your prolific, everyday prolific posters. But yeah, LinkedIn's a great place to follow and I appreciate the connect.
Jon Krohn:	00:59:48	Fantastic. Alright, well thanks for that and thanks for this whole episode again Shirish and Tyler and yeah, I feel like it's probably not going to be long. We don't have it planned, but I wouldn't be surprised if we're welcoming one or both of you on the show again in the future for more. These are the only kinds of episodes where we dig in detail on how people can be getting inference or model training on the edge, and so I always learn a lot in them. I really particularly enjoyed all the conversation today around space, space models and mixture of experts and the granite release, all really cool stuff. Thanks guys.
Tyler Cox:	01:00:29	Thank you John.

Shirish Gupta: 01:00:29 Thanks for having us again, John. Yeah, it's always a pleasure to be here and I was delighted to have Tyler join me for this episode because his technical depth is just awesome. It's all inspiring. I feel humbled every day when I enter the innovation lab where he sits and just can't get rid of my imposter syndrome, like I don't deserve to be here. This is the place where genius way beyond my years thrive. So I just enjoy working with him and his team every day.

Jon Krohn: 01:01:06 Interesting episode for sure. In it, Tyler Cox and Shirish Gupta covered state-based models like Mamba and how they use selective mechanisms to process information more efficiently than Transformers, making them ideal for edge devices where memory bandwidth is the bottleneck. They also talked about mixture of experts, architectures that activate only specific subsets of parameters for each task, allowing large models to run on resource constrained devices by keeping most parameters dormant. And they talked about how the Dell AI factory simplifies deploying and managing custom AI workloads across enterprise PC fleets by abstracting away silicon diversity, simplifying deployment and providing enterprise grade manageability, accelerating AI transformation from multi-year timelines down to weeks or months. 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 Tyler and Shirish's social media profiles, as well as my own at superdatascience.com/939.

01:02:08 Alright, that's it. Thanks to everyone on the SuperData Science podcast team, our podcast manager, Sonja Brajovic, media editor, Mario Pombo, partnerships manager, Natalie Ziajski, researcher Serg Masís, writer Dr. Zara Karschay, and our founder Kirill Eremenko. Thanks to all of them for producing another excellent episode for us today for enabling that super team to

create this free podcast for you. We're deeply grateful to our sponsors. If you're ever interested in sponsoring the show yourself, you can find out how to do that at jonkrohn.com/podcast. Otherwise, you can support us by sharing the show with people who would like to listen to it or watch it, review the show on your favorite podcasting app or on YouTube, subscribe if you're not already a subscriber. But most importantly, just keep on tuning in and 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 SuperDataScience Podcast with you very soon.