[00:00:00] Noah Kravitz: Hello, and welcome to the NVIDIA AI Podcast. I'm your host, Noah Kravitz. Our guest today is Yashraj Narang. Yash is senior research manager at NVIDIA and the head of the Seattle Robotics Lab, which I'm really excited to learn more about along with you today. Yash's work focuses on the intersection of robotics, AI, and simulation, and his team conducts fundamental and applied research across the full robotics stack, including perception, planning, control, reinforcement learning, imitation learning simulation, and vision language action models. Full robotics stack, like it says!

Prior to joining NVIDIA, Yash completed a PhD in material science and mechanical engineering from Harvard University and a master's in mechanical engineering from MIT and he's here now to talk about robots, the field of robotics, learning, all kinds of awesome stuff.
I'm so excited to have you here, Yash. So thank you for joining the podcast. Welcome. Thank you so much, Noah. So maybe first things first, and this is a very selfish question I mentioned before we started but I think the listeners will be into it too. I've never been to the Seattle Robotics Lab.

I don't know much about it. Can we start with having you talk a little bit about your own role, your background, if you like, and give us a little peek into what the Seattle Lab's all about?

[00:01:26] Yashraj Narang: Yeah, absolutely. So the Seattle Robotics Lab it started in, I believe, October of 2017. I actually joined the lab in December of 2018, and the lab was started by Dieter Fox, who's a professor at University of Washington.

At the time I believe he had a conversation with Jensen at a conference, Jensen Huang, of course the CEO of NVIDIA. And Jensen thinks way far out into the future and at that point he was getting really excited about robotics. And he said, essentially, that we need a research effort in robotics at NVIDIA.

And that's really how the lab started. So that was the birth of the lab. And at the beginning, the lab, and it still has a very academic focus. Okay. So we consistently have really high engagement at conferences. We publish a lot. We do a lot of fundamental and applied research. And recently NVIDIA has been developing, especially over the past few years, a really robust product and engineering effort as well.

So we're working more closely and closely with them to try to get some of our research out into the hands of the community. Fundamental academic mission, but it's really important for us as well to transfer our research and get it out there for everyone to use.

[00:02:32] Noah Kravitz: Fantastic. And you mentioned Dieter Fox, I believe at UDub, university of Washington is the lab, is there a relationship there?

[00:02:40] Yashraj Narang: Yeah when Dieter started the lab, we, over a number of years, had a very close relationship with University of Washington where many students from his lab and others would come do internships at the Seattle Robotics Lab. We still definitely have that kind of relationship. I stepped into the leadership role just a few months ago.

Oh wow.

[00:02:58] Noah Kravitz: Okay.

[00:02:59] Yashraj Narang: I plan to maintain that relationship because it's been so productive for us.

[00:03:03] Noah Kravitz: Awesome. I have a little bit of bias. Somebody very close to me is an alum, Go Huskies. I had to ask. Alright, let's talk about robots. We're gonna start talking about really I'll leave it to you and I'll ask at a very high level.

How do robots come to life? What does that mean when we talk about, a robot coming to life? And I think there's gonna get into the three computer concept and stuff like that, but I'll leave it to you at a high level. What does that mean, bringing the robots to life?

[00:03:28] Yashraj Narang: It's a big question. I think it's a real open question too.

I think we can even start with what is a robot. I think this is a subject of debate. Gnerally speaking, a robot is a synthetic system that can perceive the world, can plan out sequences of actions, can make changes in the world, and it can be programmed and it typically serves some purpose of automation.

That's really the essence of a robot. Now there's the question of if you have a robot, how can it come to life? So I would say that if most people were, for example, to step into a factory today, like an automotive manufacturing plant, they would see lots and lots of robots everywhere,

[00:04:08] Noah Kravitz: right?

[00:04:09] Yashraj Narang: And the motion of these robots in the payloads of these robots and the speed of these robots, it's extremely impressive. But those same people that are walking into these places, they might not feel like these robots are alive. Because they don't necessarily react to you. In fact, you probably wanna get outta their way if they're putting something together, to be safe.

So I think part of robots coming alive is really this additional aspect of intelligence so that when conditions change. It can adapt, it can be robust to perturbations and it can start to learn from experience. Yeah. And I think that's really kinda the essence of coming alive.

[00:04:47] Noah Kravitz: Got it. And what is the three computer concept and how does it relate to robotics?

[00:04:52] Yashraj Narang: Yeah, the three computer concept is is pretty interesting. I think this was, I don't know the exact history of this, but I think this was inspired by, the three body problem. So the three computer concept. It's really a formula for today's robotics. Both on the research side and the industry side, and it has three parts as the name suggests.

So the first computer is the NVIDIA DGX computer. So this includes things like GB 200 systems, Grace Blackwell, super chips, and systems that are composed of those chips. And these are really ideal for training. Large AI models and running inference on those models. So getting that fundamental understanding of the world, being able to process, take images as input language, as input and produce meaningful actions, robot actions as output, for example, training these sorts of models and then running inference on.

The second computer is Omniverse and Cosmos. It's a combination of these things. So Omniverse is really a developer platform that NVIDIA has built for a number of years with incredible capabilities on rendering, incredible capabilities on simulation, and many applications built on top of this platform.

For example, in the Seattle Robotics Lab, we're heavy users of Isaac Sim and Isaac lab which are basically robot simulation and robot learning. Software that is developed on top of Omniverse, and what you can do with Omniverse is essentially. Train robots to acquire new behaviors. For example, using processes like reinforcement learning, which is intelligent trial and error.

You can also use it to evaluate robots, for example, if you have some learned behaviors and you wanna see how it performs in a in different scenarios, you can put it into simulation and see what happens there. Cosmos is essentially a world model for robotics and world model is this kind of big term and many people have different interpretations of it.

But just to round things a bit here. Some of the things that Cosmos has done is actually make video generation models. So you could have an initial frame of an image, you can have a language command, and then you can predict sequences of image that come after that. So this is the Cosmos Predict model.

There's also the Cosmos transfer model. The idea here is that you can take an image and you can again take a, let's say a language prompt and you can transform that image to look like a completely different scene while maintaining, the shape and semantic relationships of different objects in that image.

Then there's Cosmos reason which is really a VLM, which is a vision language model. So it can take images as input language as input, and it can basically produce language as output. It can answer questions about images and it can do a sort of a step-by-step thinking or reasoning process. Now, just stepping back a little bit, second computer, again, omniverse and Cosmos, and what they're really used for is to generate data.

To generate experience and to evaluate robots in simulation. And so in a sense, this can come either before or after the first computer. You can for example, generate a lot of data and then learn from it using that first computer, these DGX systems, or you can train a model on that DGX system and then evaluate it using something like Omniverse or Cosmos.

The third computer is the A GX, by the way I looked this up recently. I was curious we've been here for a while, but still curious, what is the D and DGX stand for? What does the A and AG GX stand for? Oh, yeah. Okay. D is apparently for deep learning and a is apparently for autonomous, so it's a nice way to remember it.

So

[00:08:31] Noah Kravitz: interesting. The more you know

[00:08:33] Yashraj Narang: Yeah, exactly. The more you know, so the third computer is the Jetson, a GX system specifically. The Thor has been recently released, and this is all about. Running inference on models that are located on your robot. So instead of having, separate workstations or data centers, this is a chip that actually lives on the robot where you can, we can basically have a AI models there and you can run inference on them in real time.

Really powerful.

[00:09:02] Noah Kravitz: So before asking a follow up, I feel like I have to plug the podcast real quick, because it was really satisfying in a way that listening to you and thinking, oh yeah, we did an episode with that. Oh, yeah, Sonya talked about that. Oh, yeah. So I will say, if you would like to know a little more about the feeling of walking through an automotive factory with a lot of robots doing amazing things.

Without worrying about getting out of the way. Great episode with Siemens from a few months back. Check that out. I mentioned Sonya Fiddler Ridley recently, sorry, from NVIDIA. She spoke around siggraph, but a lot of stuff related to robots, of course from GTC. They're my plucks. Okay, so you got into this a little bit.

Yeah. But mentioning Thor in particular, but what's changed recently in the field and what does that mean for where robotics is headed?

[00:09:49] Yashraj Narang: Yeah, I think there have been many changes in the field. I think for example, the three computer solution, three computer strategy from NVIDIA, that's been definitely a key enabler.

Just the fact that there is access to more and more compute, more and more powerful compute and tools like Omniverse for example, for rendering and simulation and cosmos for world models. And of course, better and better onboard compute. I think that's really empowered robotics.

On, let's say, maybe if we think a little bit about the learning side. I think since joining the lab in December of of 2018, I've been lucky to witness different transformations in robotics over time. One thing that I witnessed early on was actually I think this was in 2019 when OpenAI released, its Rubik's cube manipulation work.

And so these are. These were basically dexterous hands, human-like hands that learned to manipulate a Rubik's cube and essentially solve it. But it was learned purely in simulation and then transferred to the real world.

[00:10:51] Noah Kravitz: Yeah, I remember that.

[00:10:52] Yashraj Narang: So that was a big moment in the rise of the sim to real paradigm training and simulation deploying in the real world.

I think other things came after that. The transformers were of course invented before, but really starting to see more and more of that model architecture and robotics. I think that was that was a big moment or big series of moments. Another specific moment that was pretty powerful was just, of course, as everybody in AI knows chat, GBT.

So I, I think that was released in late 2022. Most people started to interact with it early 2023. And then, the world of robotics started thinking about, okay, how do we actually leverage this for what we do? And, many other fields felt the same thing.

[00:11:36] Noah Kravitz: Sure.

[00:11:37] Yashraj Narang: So there was.

Really an explosion of papers starting in 2023 about how to use language models for robotics and how to use vision language models for robotics. And I think that was quite interesting. So there are papers that kind of explored this along every dimension. Can you, for example. Give some sort of long range task to a robot or in this case, to a language model and have it figure out all the steps you need to accomplish in order to perform that task.

Can you, for example, use a language model to construct rewards? So when you do, for example, reinforcement learning, intelligent trial and error, you usually need some sort of signal about. How good your attempt was. You're trying all of these different things. How good was that sequence of actions?

And that's typically called a reward. These are traditionally hand-coded things using a lot of human intuition. And there are some very interesting work, including Eureka from NVIDIA, about how to use language models to generate those rewards. There was also a simultaneous explosion in more, more general generative ai, for example, generating images and generating 3D assets.

A lot of this work came from NVIDIA as well. Yeah. So on the image generation side. There, there was work, for example, on generating images that describe the goal of your robotic system. So where do you want your robot to end up? What do you want the final product to look like? Let's generate an image from that and use that to guide the learning process.

And then there's also, when it comes to simulation. One of the, one of the, and we'll probably get more into this a little bit later, but one of the challenges of simulation is you have to build a scene and you have to build these 3D assets or meshes, and that can take a lot of time and effort and artistic ability and so on.

So there's a lot of work on automatically generating these scenes and generating these assets. And in a sense, you can view this transformation that we've seen over the past few years as taking the human or human ingenuity. More and more out of the process or at higher and higher levels as opposed to absolutely doing everything and hard coding things like rewards and final states and, building meshes and assets manually and describing scenes and so on and so forth.

So we're. Able to automate more and more of that.

[00:13:57] Noah Kravitz: There's so much in what you just said, and one of the big things for me from this perspective is thinking about how little I understood about Omniverse. Let alone cosmos before having the chance to have some of these conversations, particularly over the past few months, and having to do with robotics, physical ai and simulation in the idea of creating the world, and then the robot is able to learn and cosmos, it's all it's just.

Fascinating. It's so cool to it's, to, i'm wanting to geek out on my end. But when you're talking about the different types of learning and I'm sure they go together in the same way that you mix different approaches to anything in solving complex problems. Can you talk a little bit about, I don't know if pros and cons is the right way to describe it, but the difference between imitation and reinforcement learning?

Not so much in what they are, but in effectiveness or how you use them together and that sort of thing.

[00:14:50] Yashraj Narang: Yeah, absolutely. I think these, these are two really popular paradigms for robot learning, and I will. Try to ground it in what we do, what we typically do in robotics, the typical implementations of imitation learning and and reinforcement learning.

So in a typical imitation learning pipeline, you're typically learning from examples. So for example let's say I, I define a task, I'm trying to pick up my water bottle with a robot. What I might do if I were using an imitation learning approach is maybe, physically move around the robot and pick up the water bottle, or I might use my keyboard and mouse to teleoperate the robot and pick up the water bottle, or I might use other interfaces.

But the point is that I am collecting a number of demonstrations of this behavior. I do it once in one way. I do it, the second time in a different way, and I and maybe I move the water bottle around and I collect a lot of different demonstrations there. Basically, the purpose of imitation learning is to essentially mimic those demonstrations.

The behaviors would ideally look, as I have demonstrated it right now, reinforcement learning operates a little bit differently. Reinforcement learning tries to discover the behaviors, or the sequences of actions that achieve. The goal. So you know, in the most extreme case, what you might do if you were to take a reinforcement learning approach, again, intelligent trial and error, is you might just have proposals of different sequences of actions that are being generated.

And if they happen to pick up the water bottle, I give a reward signal of one.

[00:16:27] Noah Kravitz: Okay?

[00:16:28] Yashraj Narang: And if they fail, I might give a reward signal of zero. And the key difference here is that I am not providing very much guidance on this sequence of actions that the robot needs to use in order to accomplish the task.

I'm letting the robot explore, try out many different things, and then come up with its own strategy. Pros and cons. So imitation learning, one pro is that you can provide it a lot of guidance and the behaviors that you learn. For example if. If a human, if a person is demonstrating these behaviors, then the behaviors that you learn would generally be human-Like they're trying to essentially mimic those those demonstrations.

Now reinforcement on the other hand, again, in the most extreme case, you're not necessarily leveraging any demonstrations. The robot or agent it's often called has to figure this out on its own. And so it can be less efficient. Of course, you're not giving it that guidance, and so it's trying all of these sequences of actions and there are principled ways to do that.

But essentially it's, it would be less efficient than if you were to give it some demonstrations and say, learn from that. Now the pro is that you can often do things that you have the capability of doing things that are really, that can be really hard to demonstrate. So one of the things, one of the topics that I've worked on for some time, for example, is assembly.

Literally teaching robots to put parts together, and this can actually be really difficult to do via a teleoperation interface. You probably need to be an expert gamer in order to do that,

[00:17:55] Noah Kravitz: I hear you talk about assembling things and I think of, forget the robot. I think of myself trying to put together like very small parts on something.

Yeah. Twisting a screw in and I can't, that makes me cringe literal on trying to tell, operate a robot. Yeah.

[00:18:08] Yashraj Narang: It can be really hard depending on the task. And the second thing is that reinforcement learning generally has the potential to achieve superhuman performance. So there are things, and I think games are a great example.

One of the domains of reinforcement learning historically has been in games like Atari games. And that's where maybe in recent history got super excited about reinforcement learning because all of a sudden you could have these, aI agents that can do better at these games than any human ever.

It the same capabilities apply to robots. So you can potentially learn, the robot can learn behaviors that are better than, what any person could possibly demonstrate. And maybe a simple example of this is speed. So maybe there's a tricky problem you're trying to give your robot where it has to go through a really narrow path and has to do this very quickly.

And if you were to demonstrate this, you might proceed very slowly. You might collide along the way. But if a reinforcement learning agent is allowed to solve this problem, it could probably learn these behaviors automatically, these smooth behaviors. And it can start to do this really fast.

And, assembling objects is another example. You can start to assemble objects faster than you could possibly demonstrate. And I think that's the power.

[00:19:19] Noah Kravitz: That's very cool. The thinking about or listening to you talk about different approaches to teaching and learning brought to mind. I was looking at the NVIDIA YouTube channel just the other day for a totally different reason, and came across the video of Jensen giving the robot a gift and writing the card that says, dear Robot, enjoy your new brain.

Or something along those lines. There's something I only know the name modular versus end-to-end brain. What is that about? Is that, am I along the right lines or is that something totally different?

[00:19:46] Yashraj Narang: No, that's it's essentially a way to design robotic intelligence. I would say these are two competing paradigms.

Both of these paradigms can leverage the latest and greatest in hardware. I would say that now the modular approach is an approach that has been developed for a very long time in robotics. And the, a classic framing for this is that a robot, in order to perform some tasks or set of tasks, needs to have the ability to perceive the world.

So to take in sensing information and then come up with an understanding of the world, like where everything is, for example. And it also needs the ability to plan. For example, given some sort of model of the world like a physics model for example, or a more abstract model and maybe some sort of reward signal.

Can it actually select a sequence of actions that is likely to accomplish a desired goal?

And then, a third module in this modular approach would be the action module. And that means that, okay, you get in this sequence of actions, maybe this, these configurations that you'd like the robot to reach.

In space and the action module would figure out, also called control, would figure out what are the motor commands that you wanna generate? Literally what are the signals you wanna send to the robot's motors in order for it to, move along this path in space. So that's the perceived Plan Act framework.

It's called different things over time, but. That's the classic framing for a modular approach. And so following that you would have maybe a perception module and you'd have some group of people working on that. You'd have a planning module, you'd have some group of people working on that.

You'd have an action module. And this is how re many robotic systems have been built over time. Now, the end-to-end approach is something that is is definitely newer. And the idea is that you don't draw these boundaries really you you take in your sensor data like camera data, maybe force tort data if you're interacting with the world and then you directly predict.

The commands that you may send your motors, right? So you skip these intermediate steps and you go straight from inputs to outputs, and that's the end-to-end approach. And, I would say the modular approaches are extremely powerful. They have their advantages, which there are real there's a lot of maturity around developing each of those modules can be easy to debug, for teams of engineers, which I was mentioning the groups of people earlier.

It can be easier to certify as well. If safety is a safety critical application, the end-to-end approach the advantage there is that you're not relying as much on human ingenuity or human engineering to figure out what exactly are the outputs I should be producing from my perception module.

What exactly are the outputs I should be producing for my planning module and so on. That requires a lot of engineering and if. If you don't do it right, you may not pick the desired outcome.

[00:22:35] Noah Kravitz: Yeah, I was just gonna say, conceptually, it made me think of the difference between doing whatever task I'm used to doing and asking a chat bot just to shoot me the output.

And, yeah. Yeah.

[00:22:46] Yashraj Narang: And I think just another analogy here would be, I think this has been a really fruitful debate. Really vigorous debate in autonomous driving actually. So in the 2010s, I would say just about every effort in autonomous driving was focused on the modular paradigm.

Again, separate perception, planning, control modules and different teams associated with each of those things. And then late let's say, early in the 2020s was a real shift to the end-to-end paradigm, which basically said, let's just collect a lot of data and train a model that goes directly from pixels to actions.

Actions in this case being steering angle, throttle breaks, and so on. Yeah. And many things today look. I would say like a hybrid. Different companies have different strategies, but most people have converged upon something that has elements of both.

[00:23:37] Noah Kravitz: I'm speaking with Raj Nang Yas is a senior research manager at NVIDIA and the head of the Seattle Robotics Lab, and we've been talking about all things we robots, AI simulation, which we'll get back to in a second.

But we were just talking about different styles, different approaches to robotics learning. Wanted to go back to earlier in the conversation when you mentioned, going into the factory and seeing all these different robots doing these kinds of things, and even before that, your definition of what a robot is or is not.

And thinking about that and I'm getting to thinking about. Asking you to define sort of the difference between traditional and humanoid robots. And I'm thinking traditional like robot arms in a factory. I have fuzzy probably images from sci-fi movies when I was a kid and stuff like that.

And humanoid robots, and I mentioned this earlier, back during GTCI had the chance to sit down with the CEO of one X robotics, who we talked all about human robots. So maybe you can talk a little bit. About this traditional robots, humanoid robots, what the difference is, and maybe why we're now starting to see more robots that look like humans and whether or not that has anything to do with functionality.

[00:24:51] Yashraj Narang: Yeah, absolutely. So one of your earlier questions too is how has, how has robotics changed recently? Yeah. And I think this is just a. Another fantastic example of that. It's been unbelievable over the past few years to see the explosion of interest and progress in humanoid robotics. And, to be fair actually, companies like Boston Dynamics and and Agility Robotics, for example, have been working on this since, probably the mid, maybe even early 2010s.

[00:25:19] Noah Kravitz: Yeah.

[00:25:20] Yashraj Narang: And so they made continuous progress on that and everybody was always really, excited and inspired to see their demo videos and so on.

[00:25:28] Noah Kravitz: Can I interrupt you to ask a really silly question, but now I need to know of, is there a word we say humanoid robots, right?

Is there a word for a robot that looks like a dog? Because the Boston Dynamics makes me think of those early Atlas, I think those early videos.

[00:25:41] Yashraj Narang: Yeah. Yeah. And I think Boston Dynamics used to they, they had a dog-like robot, which was called Big Dog. Oh, okay. Yeah.

Sometime then, which is, maybe why it's just come to mind. People typically refer to them as quad eds. Quads. Okay. Four, four legs. Got it. Thank you. No problem. Yeah. Yeah, where were we? Traditional robots versus humanoids. There's been explosion of interest in humanoids, particularly over the past few years, and I think it was just this perfect storm of factors where there was.

Already a lot of excitement being generated by some of the original players in this field. Folks like Tesla got super interested in humanoid robotics, I think 20 22, 20 23. And it also coincided with this explosion of advancement in intelligence through LLMs VMs and early signals of that in robotics.

And so I think, there's a group of people, forward-thinking people Jensen very much included. This is near and dear to his heart that felt that that time is right for this dream of humanoid robotics to finally be realized. Let's actually go for it. And, this begs the question of why humanoids at all?

Why have people been so interested in humanoids? Why do people believe in humanoids? And I think the most common answer you'll get to this. Which I believe makes a lot of sense is that the world has been designed for humans. We have built everything for us, for our form factors, for our hands, and if we want robots to operate alongside us in places that we go to every day, in our home, in the office and so on.

We want these robots to, to have our form. In doing so, they can do a lot of things ideally that we can go up and down stairs that were really built for the dimensions of our legs. We can open and close doors that are located at a certain height and have a certain geometry because they're easy for us to grab.

Humanoids could, manipulate tools like hammers and scissors and screwdrivers and pipettes if you're in a lab. These sorts of things, which were built for our hints. And so that's really the fundamental argument about why humanoids at all. And it's been amazing to see this.

Iterative process where there's advancements in intelligence and advancements in, in the hardware. So basically the body and the brain and going back and forth and just seeing, for example, the amount of progress that's in that's been happening over the past couple of years in developing really high quality robotic hand hardware.

It's amazing. So that's really my understanding of the story and the fundamental argument behind behind humanoid robots.

But I definitely see. I would say I see a future where these things actually just coexist, traditional and humanoid. Yeah.

[00:28:10] Noah Kravitz: So earlier we were talking about the importance of simulation, creating world environments where robots can explore, can learn all the different approaches to that.

And I think we touched on this a little bit, but can you speak specifically to the role of simulated or synthetic data versus real world data? It's something we touched upon, and again, listeners, the more we're talking about. I feel like all these recent episodes coming together, talking about, the increasing role of AI broadly.

Generating tokens for other parts of the system to use and all of that. So when it comes to the world of robotics, simulated data, real world data, how do they work? How do they coexist?

[00:28:50] Yashraj Narang: Yeah, so first I, I'd like to say that in contrast with a number of other areas like language and vision. Robotics is widely acknowledged to have a data problem.

So there is no internet scale corpus of robotics data. And so that's really why so many people in robotics are very interested in simulation and specifically using it to generate synthetic data. So that's basically the idea is that simulation can be used to have high fidelity renderings of the world.

They can be used to do really high quality physics simulations. They can be used as a result to generate a lot of data that would just be totally intractable to collect in the real world and real world data is, generally speaking your source of ground truth. It doesn't have any gap with respect to the real world because it is the real world, but it tends to be much harder to scale.

In contrast with autonomous vehicles, for example, robotics doesn't really have a car at the moment. There aren't fleets of robots that everybody has access to.

[00:29:50] Noah Kravitz: Can't put a dash cam on the those little food delivery robots. Yeah. And get the data you need.

[00:29:54] Yashraj Narang: Yeah. And even if you could, what would be nearly enough data? The answer is probably no. To, to train general intelligence. That's why people are really attracted to the idea of using simulation to, to generate data and real world. Whenever you can get it, it's the ideal source of data. But it's just really difficult to scale.

[00:30:13] Noah Kravitz: So you mentioned, using real world data, there's no gap. We've talked about the sim to real gap in other contexts. How do you close it in robotics? What's the importance of it? Where are we at? And you talked about it a little bit, but get into the gap a little more and what we can do about it.

[00:30:30] Yashraj Narang: Sure. So SIM to real gap. So there are different areas in which simulation is typically different from the real world. So one is, on the perception side.

Literally, the visual qualities of simulation are very different from the real world. Simulation looks different often from the way the real world does.

So that's one source of gap. Another source of gap is really on the physics side. For example, in the real world, you might be, trying to, manipulate something, pick up something that is very flexible and your simulator might only be able to model rigid objects, or rigid objects connected by joints.

And even if you had a perfect model in your simulator of whatever you're trying to move around or manipulate. You still have to figure out like what are the parameters of that model? What is the stiffness of this thing that I'm trying to move around? What is the mass what are the inertia matrices in these properties?

So physics is just another gap. And then there are other factors, things like latencies. So in the real world you might have different sensors that are streaming data, different frequencies. In simulation, you may not have modeled all of the complexities of different, again, different sensors coming into different frequencies.

Your control loop may be running at a particular frequency and these things may have a certain amount of jitter or delay in the real world, which you may or may not model in simulation.

[00:31:50] Noah Kravitz: Okay.

[00:31:50] Yashraj Narang: So these are just a few examples of areas where you. It might be quite different between simulation in the real world and generally speaking, the ways around this are you either spend a lot of time modeling the real world, really capturing.

The visual qualities and the physics phenomena and the physics parameters and the latencies and putting that in simulation. But that can take a lot of time and effort. Another approach is, called domain randomization or dynamics randomization, and the idea is that you can't possibly identify everything.

About the real world, right? And put it into simulation. So whenever I'm doing learning on simulated data, lemme just randomize a lot of these properties. So I want to train a robot that can pick up a mug. Or, put two parts together and it should work in any environment. It shouldn't really matter what the background looks like.

So let me just take my simulated data and randomize the background in many different ways. And you can do similar strategies for physics models as well. You can randomize different parameters of physics models. And then there's also another approach which is really focused on. Domain adaptation.

So I really care about a particular environment in which I wanna deploy my robot. So let me just augment my simulated data to be reflective of that environment, right? Let me make my simulation look like an industrial work cell, or let me make it look like my home because I know I'm gonna have my robot operate here.

And maybe the final approach is this thing called. Domain in variance. So there's randomization, adaptation, and in variance which is basically the idea that I'm gonna remove a lot of information that is just not necessary for learning. If maybe if I'm picking up certain objects, I only need to know about the edges of these objects.

I don't need to know what color, for example. Yeah. Taking that idea. And incorporating it into the learning process and making sure that my, my networks themselves or my data might be transformed in a way that it's no longer reliant on these things that don't matter.

[00:33:50] Noah Kravitz: Yeah. Thinking about all of the data coming in and all the things that, that can be captured by the sensors and using video to train.

And earlier you were talking about the problem and it made me think of reasoning models, the problem of, can you give a robot a task and can it break it down and reason its way and then actually execute and do it. What are reasoning VLA models been talked about? Not recently. I keep hearing about them anyway.

Can you talk a little bit about what they are and how they're used in robotics?

[00:34:19] Yashraj Narang: Absolutely. So reasoning itself, just stepping back for a second, reasoning is an interesting term because it means many things to many different people.

[00:34:26] Noah Kravitz: Yeah.

[00:34:26] Yashraj Narang: I think a lot of people think about things like logic and causality and common sense and so on, as different types of reasoning, and you can use those to draw conclusions about the world.

Reasoning in the context of LLMs and VMs and now VLAs. So vision, language, action models that produce actions as outputs often means, in, in simple terms, thinking step by step. In fact, if you go to chat GBT and you say, here's my question, show me your work or think step by step, it will do this form of reasoning.

And so that's the idea is that you can often have. Better quality answers or better quality training data if you allow these models to actually engage in a multi-step thinking process. And that's the essence of reasoning models and reasoning VLAs are no exception to that. Okay, so I might give a robot a really hard task like setting a table and maybe I want my VLA to now identify what are all the subtasks involved in order to do that.

And within those subtasks, what are all the smaller scale? Trajectories that I need to generate and so on. So this is the essence of the reasoning VLA.

[00:35:31] Noah Kravitz: Got it. So to start to wrap up here, I was going to ask, I am going to ask you to in a way, it's summarizing what we've been talking about, but maybe to put a point on what you think the.

Most important current limitations are to robotic learning that, we're working, you're working, you and your teams and folks in the community are working to overcome. You mentioning setting the table though made me think, a better way to ask that. How far are we from laundry folding robots?

Am I gonna I'm the worst at folding laundry and I always see demos and I heard at some point that, folding laundry represents conceptually a very difficult. Task for a robot. Am I gonna see it soon before my kids go off to school? And,

[00:36:14] Yashraj Narang: I think you might see it soon.

I've seen very cool some really impressive work coming out recently, from various companies and demos within NVIDIA on things like laundry folding. And the general process that people take is to collect a lot of demonstrations of people actually folding laundry and then use imitation learning paradigms or variance paradigms.

Try to learn from those demonstrations. And this ends up actually being, if you. The right kind of data and enough data in the right model architectures, you can actually learn to do these things quite well. Now the classic question is, how well will it generalize if I learn to fold? If I have a robot that can fold my laundry, can it fold your laundry?

The typical answer to that is you probably need some amount of data. That's in the setting that you actually want to do global robot in. And then you can fine tune these models, but I would say we're actually pretty, we're getting closer and closer, closer than certainly I've ever seen on tasks like laundry folding.

[00:37:16] Noah Kravitz: I'm excited. I'm excited. That's you've got me optimistic and I thank you for that. So perhaps to get back to the more general Yeah. Conversation of interest, the current limitations, what do you see them as? And what's the prognosis on, on. Getting past them.

[00:37:31] Yashraj Narang: Sure. I think one big one is people feel, I would say the community as a whole is really optimistic about the role of simulation, robotics, or at least most of the community. It simulation can take different forms. It can take the physics simulation approach, or it can take this, video generation.

Let me just predict what the world will look like. And these are really, really thriving paradigms. And I think two questions around that. One that we just talked about, which is the sim to real gap. So I think sim to real gap is people have made a lot of progress on, it's something we've worked very hard on NVIDIA, but there's still a lot more progress to be made.

Until we can truly. Generate data and experience and simulation and have it transfer to the real world without having to, put a lot of thought and engineering into truly making it work. And conversely there's the real toim question. Building simulators is really difficult.

You, again, have to, design your scenes and design your 3D assets and so on. Wouldn't it be great if we could just take some images or take some videos of the real world and instantly have a simulation. That also has physics properties. It doesn't just have the visual representation in the world, but it has realistic masses and friction in, in, in these other properties.

So sim to real and real to sim, I think are two big challenges and we're just getting closer and closer every few months on, on solving those problems. And then the boundaries between SIM and REAL, I think will start to be a little bit blurred which is maybe an interesting possibility.

I think that's one big thing. And the second big thing I'd say for now is the data question again, robotics, as we're talking about it here, doesn't have the equivalent of a car. There is no fleet of robots that everybody has access to that can be used to collect a ton of data. Until that exists, I think we have to think a lot more about where we're gonna get that data from.

And one thing that the grit effort at NVIDIA, which is around humanoids, has proposed, is this idea of the data pyramid. Where you basically have, at the base of the pyramid, things like videos, YouTube videos that you're trying to learn from. And then maybe a little bit higher in the pyramid.

You have things like synthetic data that's coming from different types of simulators, and then maybe at the top of the pyramid you have something like. Data that's actually collected in the real world. And then the question is, what is the right mixture of these different data sources to give robots this, general intelligence.

[00:39:48] Noah Kravitz: Yash, as we're recording this coral is coming up. Let's end on that forward looking note, and it'll be a good segue for the audience to go check out what Coral's all about, but tell us what it's about and what your NVIDIA's participation is gonna be like this year.

[00:40:01] Yashraj Narang: Yeah, absolutely. Coral is stands for the Conference on Robot Learning.

And it started out as a small conference, I think in, it was probably 2017 was maybe the first edition of it. And it's grown tremendously. It's one of the hottest conferences in robotics research now as learning itself has a paradigm, has really taken off. This year it's gonna be in in Seoul, in Korea, which is extremely exciting.

[00:40:23] Noah Kravitz: Yeah.

[00:40:24] Yashraj Narang: And it's gonna bring together the robotics community, the learning community, and the intersection of those two communities. And I think everybody in robotics is looking forward to this. Our participation, the Seattle Robotics Lab and other research efforts at NVIDIA, for example the Gear Lab, which focuses on humanoids, presenting a wide range of papers.

And so we're gonna be giving talks on those papers, presenting posters on those papers, hopefully some demos. And, we're just gonna be really excited to talk with with researchers and people would be interested in joining us in our missions.

[00:40:53] Noah Kravitz: Fantastic. Any of those posters and papers you're excited about in particular?

Maybe you wanna share a little teaser with us?

[00:41:00] Yashraj Narang: Yeah I'm excited about a number of them, but one that I can just call out for now that I work closely on is this project called Neural Robot Dynamics. That's the name of the paper. Okay. And we have, abbreviated that to Nerd

[00:41:12] Noah Kravitz: I was gonna ask.

I'm glad. Yeah.

[00:41:15] Yashraj Narang: So it's, yeah, it's just that. NE rd also inspired by neural radiance fields.

[00:41:19] Noah Kravitz: Of course. Yeah.

[00:41:20] Yashraj Narang: So we have this framework and these models which we call nerd. And the idea is basically that classical simulation, so typical physics simulators work in this way where they are, performing these explicit computations about, here are my.

Joint torques of the robot. Here's some external forces, here's some contact forces. And let's predict the next state of the robot. And the idea behind neural simulation is, can we capture that all with a neural network? And so the, you might be wondering why would you want to do that? And there is some some advantages to this.

So one is that, neural networks are inherently differentiable. What that means is that you can understand if you slightly change the inputs to your simulator, what would be the change in the outputs? And if you know this then you can perform optimization. You can figure out how do I optimize my inputs to get the robot to do something interesting, essentially, right?

Neural networks are inherently different. And if you can capture. A simulator in this way you can essentially create a differential simulator for fruit, which is which is exciting.

Another thing which is really exciting to us is fine tune ability. So it's very difficult if you're given a simulator and you want, and you have some set of real world data that you collected on that particular robot that you're simulating to actually figure out how should I modify the simulator to.

Better predict that real world data and neural simulators can do this very naturally. You can fine tune them just like any other neural network. So I can train a neural network on some simulated data and then collect some amount of real world data and then fine tune it. And this process can be continuous.

If my robot changes over time or there's wear and tear, I can continue fine tuning it and always have this really accurate, simulator of that robot, which is pretty exciting. Yeah,

[00:43:11] Noah Kravitz: that's really cool.

[00:43:12] Yashraj Narang: Yeah, I think it's really cool. And a third advantage, which we are in the early stages of exploring is really on the speed side.

A lot of compute today as many people know, it's been really optimized for AI workloads and specific types of mathematical operations, specific types of matrix multiplication, for example, that are very common in neural networks. And if you can. Transform a typical simulator into a neural network, then you can really take advantage of all of these speed benefits that come with the latest compute and with the latest software built on top of that.

So that's really exciting to us. And we did this project in a way. That allows these neural models to really generalize. So for given a particular robot, if you put it in a new place, you know in the world or you change some aspects of the world, this model can still make accurate predictions and it can make accurate predictions over a long time scale.

[00:44:06] Noah Kravitz: Amazing. For listeners who would like to follow the progress at Coral, in particular Seattle Robotics Lab in particular, NVIDIA more broadly where are some online places? Some resources you might direct them to?

[00:44:20] Yashraj Narang: Yeah, I'd say the Coral website itself is probably your your primary source of information.

So you'll find the program for Coral. You'll find links to actually watch some of the toxic coral. You'll be able to have links to papers and you'll see the range of workshops that are gonna be there. And a lot of them, I'm sure will post recordings of these workshops. That's a great way to get involved.

[00:44:41] Noah Kravitz: And that's just crl.org for the listeners?

[00:44:44] Yashraj Narang: Yes. Yes, that's right. Yeah. You get your website as well. I'm sure we'll have on the website and through NVIDIA social media accounts. Noah, you could probably call out to those. I'm sure there's gonna be plenty of updates on Coral over the next fantastic.

Yeah.

[00:44:57] Noah Kravitz: Next period of time. Can I ask you as a parting shot here? Predict the future for us. What does the future of robotics look like? You can look out a couple of years, five years, 10 years, whatever timeframe makes the most sense. And, we wanna hold you to this but what do you think about when you think about the future of all this?

[00:45:13] Yashraj Narang: Yeah, I think it comes down to those fundamental questions. One is what will the bodies of robots look like? So this is what you touched on with robot arms and factories versus humanoids. And I think what you'll see is that there'll be a place for both. Robot arms and more traditional looking robots will still operate in environments that are really built for them or need an extremely high degree of optimality.

And humanoids will really operate in environments where they need to actually be. Alongside humans and, in your household and in your office and so on, around many things that have been built for humans. So I see that as the future of the body side of things. On the brain side of things, there's also these questions of, modular versus end-to-end paradigms.

And what I've seen in autonomous vehicles is, of course, as we talked about before, starting with modular swinging to end-to-end. Starting to converge on something in the middle. And I can imagine that robotics as we're talking about here, for example, robotic manipulation, will start to follow a similar trajectory where we will explore end-to-end models and then probably converge upon hybrid architectures until we collect.

Enough data that an end-to-end model is actually all we need. That's how I see those aspects. There's some other questions. For example, are we gonna have specialized models or are we just gonna have one big model that solves everything?

That one is a little bit hard to predict, but I would say that again, there's probably a role for both where we're gonna have specialized models for very specific domain specific tasks and where, for example.

Power or energy limits are very significant and you're gonna have sort of these generalist models in other domains where you need to do a lot of different things and you need a lot of common sense reasoning to solve tasks. Yeah, I would say those are some open debates and that would be my prediction.

And then maybe one other thing that you touched on was simulation versus the real world. And again, I see. This is one of the most exciting things. I'd love to see how this unfolds, but I really feel that the boundaries between simulation and real world will start to be blurted. The sim to real problem will be more and more solved, and the real to sim problem will also be more and more solved.

And so we'll be able to capture the complexity of the real world and make predictions in a very fluid way. Perhaps using a combination of physics simulators and these world models that people have been building like cosmos.

[00:47:35] Noah Kravitz: Amazing future Yash. Thank you so much. This has been an absolute pleasure and I know you have plenty to get back to, so we appreciate you taking the time out to come on the podcast.

All the best with everything and enjoy Coral. Can't wait to follow your progress and read all about it. Thank you so much, Noah. It's been a pleasure.