**Vincent:** 0:00

RT2 is a multimodal model, so very similar to Gemini in spirit, and it can understand, taken as an input of language and images, and we've trained it on robotics data, which is paired data of you know, images and actions, to basically be able to not just speak English but also speak Robotics, if you will. Basically, we treat robot actions as merely another language that this chatbot can speak, and that has a lot of benefits from an architectural standpoint. That means the model can take advantage of common sense, understanding that a large language model has A large language model knows that you can put a cup on the table, but putting a table on the cup doesn't make much sense.

**Craig:** 1:00

Hi, I'm Craig Smith and this is Eye on AI. Household robots are getting closer to reality, and in this episode I speak with Vincent Van Hoek, a senior researcher at Google's DeepMind, about their work on robotic control, using large language models and visual language models to create something that they call visual action models. We discuss the architecture of their robotic transformer system, rt2, which combines visual perception, natural language understanding and robotic action planning into a single multimodal transformer model. Vincent explains how this approach allows for common sense reasoning, generalisation across different robotic platforms and the integration of world models for future action planning. I hope you find the conversation as amazing as I do. Ai might be the most important new computer technology ever. It's storming every industry and literally billions of dollars are being invested, so buckle up. The problem is that AI needs a lot of speed and processing power. So how do you compete without cost spiralling out of control? It's time to upgrade to the next generation of the cloud Oracle Cloud Infrastructure or OCI. Oci is a single platform for your infrastructure, database application development and AI needs. Oci has four to eight times the bandwidth of other clouds, offers one consistent price instead of variable regional pricing and, of course, nobody does data better than Oracle. So now you can train your AI models at twice the speed and less than half the cost of other clouds. If you want to do more and spend less like Uber, 8x8, databricks, mosaic take a free test drive of OCI at oracle.com. Slash IonAI that's E-Y-E-O-N-A-I all run together. Well, Vincent, I'm really, really delighted that you could join me. I have been talking to people about AI control of robotics for quite a while. I talked to Sergey Levine and Peter Abiel, and I've spoken with Alex Kendall at Wave, who's working with world models. But I was particularly taken by the RT2 visual language action model. I wanted to understand that and understand how it compares to things that have gone before. But to start, could you introduce yourself, give your educational background, how you got to DeepMind? I know that you've only been on the robotics team for a year or so.

**Vincent:** 4:15

Not quite, not quite. I can give you a bit of a background. I've been at Google for about 16 years now. I started working on speech recognition originally and then caught the sort of deep learning bug and started working on all areas of AI and machine learning, worked a lot on computer vision and then asked myself at some point to imagine a future in which computer vision really, really works well? What are the consequences for the world? It became very clear to me that the main consequence was that robotics would be changed forever and that actual robots operating in the real world would be a possibility. About seven years ago I pivoted to exclusively focusing on robotics and I founded the robotics effort in Google Brain at Google and started sort of developing. At the time it was really a niche research agenda. How do you use deep learning and machine learning, the modern versions of machine learning, to control robots? It was not an obvious path at the time. By now it's a lot more well-established and there are a lot of people working on the topic, the last iteration of which is how do we use the latest kind of revolution in AI, the large model revolution, large language models or large multimodal models to benefit robotics?

**Craig:** 6:04

Right. Rt2 is a robotics transformer. These are transformer models. My understanding is it's a combination of a large language model, a visual language model and reinforcement learning. You talk about reward and thumbs up and thumbs down and that sort of thing. Can you walk us through that architecture and which are the new elements the large model is coming from?

**Vincent:** 6:43

yeah, the simplest version of it, in fact, is RT2 is a multimodal model, so very similar to Gemini in spirit, and it can understand, take in as an input both language and images, and we've trained it on robotics data, which is paired data of images and actions, to basically be able to not just speak English but also speak robotese, if you will. Basically, we treat robot actions as merely another language that this chatbot can speak, and that has a lot of benefits from an architectural standpoint. That means the model can take advantage of the common sense, understanding that a large language model has. Large language model knows that you can put a cup on the table, but putting a table on the cup doesn't make much sense, or that if you have an open bottle it might spill, but if it's closed you're probably fine to pick it up. And it also has a very good sense of how to perceive the world at large. It's trained on a lot of internet images and so it has semantic information about everything that can be perceived in the world. And what we've seen is that, by treating this as a unified model that does perception, semantic understanding and action all at once, we are able to transfer those capabilities between the perception model and the actuation, and so suddenly the robots understand objects that it's never seen before. It understands concepts of making coffee that it never has encountered before, in a way that transfers very well to the actuation. In RT2, there is actually no reinforcement learning involved. It's entirely trained using supervised data, so very similar to how you would train a large language model, essentially.

**Craig:** 9:06

Yeah, the reasoning. You need a fairly I'm not sure technical term, but you need a fairly long chain of reasoning for it to plan. Large language models themselves don't plan very well, so where does the planning come in and how do you refer back?

**Vincent:** 9:37

One thing that's very interesting is that we can uplift the planning to happen essentially in semantic space. So if you want to make a plan for a robot, if you, for example, want your robot to make you a cup of coffee, the number of steps involved is very long. You have to drive your robot in certain directions and then move the arm. It's got lots of degrees of freedom. That needs to get right. So you would think that it's a very long planning problem. But if you think about the problem at the semantic level, how do I make coffee? You can ask your favourite chatbot and prompt it and say I'm a robot, how would I go about making coffee? It will give you some steps that are very plausible, which is to find the coffee machine, put the cup in the coffee machine, press the button. That's actually a very compact representation of what it means to make coffee, and so you can take that, use it as a semantic planner, like you would be, as if you were talking, walking a person through the steps required to the task and then, for each of the step, then you can decompose that into actual actions, and that's what RT2 does is taking this kind of low level sort of drive to the coffee machine and turn that into a low level action that is understandable by the robot controller.

**Craig:** 11:10

Essentially, and those actions when it translates it into actions, because it's a transformer model, it's breaking the actions into tokens. Is that right?

**Vincent:** 11:27

no-transcript. Yeah, so each token represents basically a configuration of the action space of your robots. So it's basically literally the joint angles of your robots or the position of the gripper of the robot, or you can imagine different ways of representing that action space, but at the end of the day you can think of it as a program that instructs the robots. And those language models are really good at coding. We've seen that they're very good at producing sensible code. So this is just another instance of very specific code that's tailored to a robot, that gets produced by the language model and can get translated into actions. And we do that in a closed loop so that every time the robot does an action we check the state of the world after that action and we'll replan and re-decide on what the next steps are.

**Craig:** 12:33

Yeah, well, that's where I thought reinforcement learning might come in. I saw you were talking to Tech Crunch and you talked about giving the model a reward. Maybe I'm just confused.

**Vincent:** 12:51

Well, no, no, it's totally correct, you can do that on top of it. So in the case of RT2, we did not do much reinforcement learning, but the next natural step is indeed having a reward, basically checking what the robot does, giving it a reward based on whether it succeeded or not, and feeding that information back to the model. In the general world of large language models, you have supervised learning, and then you have RLHF on top of it. It's the same kind of dichotomy where you first fine-tune the model and then, on top of that, you use a reward model to make it even better at the

specific tasks that you care about.

**Craig:** 13:37

And how do you get around hallucinations? Because in both the VLM and the LLM hallucinations you can suppress them to a point through fine-tuning, but you can't eliminate them. For example, the tokens you were talking about the specific angles in the actuators in the robot arm, for example are those encoded in or embedded in a vector database, so that the models are retrieving something very specific rather than generating them from their weights.

**Vincent:** 14:28

No, but one thing that the robot has that a typical LLM doesn't have is it has the real world in front of it. So if your robot is in your kitchen, it sees your kitchen. It doesn't hallucinate in the kitchen, it's right there. The real world is really there to ground the actions, and it makes it a lot harder for it to hallucinate objects that would not exist. It's still technically possible that it will hallucinate that there is a cup of coffee on your table when there is not. But we have this constant reinforcement of the robot seeing the environment it's in and repeatedly getting feedback about what is there, what are the consequences of its actions, and so it really mitigates this issue that other models have that are not grounded in any specific real world context. Where you actually have an anchor point that it's the real world. You cannot change that, you cannot hallucinate that Its reality is the reality check, in a sense, for the robot.

**Craig:** 15:47

Yeah, and that's why I've been interested in world models in Yanlacoon's work and Alex Kendall's work, because it's unlike LLMs that are grounded in knowledge encoded in text, which is one layer removed from reality. They perceive reality directly and then create a representation of that reality and then do planning at the representation level. Does that happen in your model? How does it differ from, for Gaia 1, with Waves AI, for example? My understanding is it's got input data coming from its cameras, it creates a representation and then it can plan in the

representation space and then there's a decoding to a control mechanism.

**Vincent:** 16:55

And in your case, In the context of self-driving cars, the world can be sort of, and I don't know what wave does specifically, but in general it's a geometric problem where you have to avoid collision with pedestrians. You have the rules of the road, so you have a. You can distil the world into a geometric representation and plan in that geometric representation. In robotics at large, it tends to be a lot harder to come up with the simplified or distilled representation of the world in which you can plan, and so that's why we've kind of focused more on the semantic planning, right. So we're trying again, taking the example of making a cup of coffee, I'm going to formulate a plan of how to make this cup of coffee, not as a geometric plan, like go to XYZ, coordinate, but as a semantic plan, like go to the coffee machine and then let the low level system understand this. But there is a place for world models and in fact the fact that we now have very good generative AI opens up the possibility of creating generative world models that enable basically the robots to imagine the future, conditioned on the actions that they're taking. Right. So you could imagine, if you have a perfect video generative model, you can take your robots and say, imagine the future if I press this button or if I go in this place and have the video generative model generate an imagined future of what could happen if you do this and in fact we've done some experiments where we do generate this futures over some certain horizon and then look at the outcome and decide is this a better outcome or is this closer to our goal or not, and use that as a way to guide the planning process. It's literally the robot is dreaming of its future and or possible futures, depending on which actions get taken, and plans using that imagined future to make its decision. I think we're gonna see a lot more of this in the future.

**Craig:** 19:29

Right, and that would be then combining a world model with the VLA. Is that right? For yeah, and just forgive me, I'm a journalist, but that's part of the role is to bring it down to layman's understanding. So you've got a robot with a camera and touch sensors and other things. The visual and all of the sensory data comes in, and then the VLM interprets that and recognizes the objects and recognizes the robot's position in that and encodes that in tokens and then the LLM is the interface with the human.

**Vincent:** 20:32

The LLM is the interface with the human, but it's a lot more than that, and that's one thing that is fascinating about those large language models is that you think that they're about language, but they're really about a lot of. It is about common sense reasoning. They have a lot of information about the world. They know about the human world indirectly, essentially through all reading all the textual data that's available on the web, and that common sense reasoning was one of the hardest things to get access to or to be able to manipulate in AI in general and for robotics in particular. Understanding this common sense is really really critical if you wanna do any form of planning in a real world environment. So the LLM is really this repository of knowledge about the human world that the robot can draw from and plan against. So it's not just language, it's also about understanding, and that's what makes it very exciting for me, what makes it very exciting from that standpoint.

**Craig:** 21:50

Yeah, so when you give the system the VLA in instruction in natural language, the LLM breaks it down into sub-task to execute that language. And then what's the length then to the VLM, to telling the robot how to move its actuators to affect the action?

**Vincent:** 22:23

The VLM and the LLMs are all connected through those token representations, and so the vision, if you will, is also an input to the system. And you take an image and you turn it into a stream of tokens. You take the command from the person and take another stream of tokens. In fact, you just concatenate them all into one long string of things and then the transformer takes those tokens and turns them into tokens that are in robot ease essentially. That, basically, are action tokens that are translated by the hardware into actions. So it's all trained together. It's all one big model, one transform, big transformer model that handles both the perception, the reasoning and the action at the same time. And the fact that this is a joint model really means that the robot can reason about the semantics of things, the vision aspects and its own embodiment, its own proprioception, if you will, in one shot, very much like what humans would do. You don't necessarily segregate your understanding of the vision from your understanding of the semantics of the world. It's one common model and what's interesting is that that model can be trained on text data, image data from the web, robot action data from our robot experiences, and all of that gets merged into something that is more than the sum of its parts, if you will.

**Craig:** 24:08

Yeah, and the point of this is to create a robot brain, so to speak, that can generalise in eventually almost any environment or situation. You've collected a lot of data globally from different labs. I was talking to Sergey and then Peter Chen at Coop-Ary and they were talking about networking with robots in different form factors and different environments globally and all of that data coming back to train a model. Are you doing that?

**Vincent:** 24:57

Yeah, so this is this RTX project. One thing that was very surprising when we started working on this RT2 project is that we found that you could imagine that every robot is different, has different embodiments, different degrees of freedom or different configuration, and that you would want to train a model per robot because they're very different. What we found is that the different robot languages, if you will, that you had to produce are mere dialects of each other, and that you don't really need to train a separate model for every robot, and that if you train a model that goes across robots, you actually get some benefit from each of them. Like there is positive transfer between the different robots. So you add one additional robot to the mix with its own experiences, its own data, its own embodiment, and everybody benefits. And so, with the RTX project, we try to stress that hypothesis by going to 20 plus different institutions asking them hey, let's pull together all our data and don't curate anything, just take the data you have. It's not. The cameras are in different positions, that's okay. The robots are different, that's okay. Your tasks have nothing to do with each other, it's okay. Just pull everything together, let's train a big model on all this data, and then we send the model back to the institutions and ask them to evaluate it so that the evaluation would be fair and we wouldn't have a hand in it and they got uniform gains from using this bigger, more general model. So this is very exciting for everybody in robotics because in computer vision we saw that collecting large amounts of data from very various sources really improved the vision model. It wasn't obvious that you could do that in the context of robotics. Now we have at least one proof point that pulling data and resources together could improve things. I think that will change how people think about robotics data. It will enable more collaboration between the different labs. I think it will have a very positive virtuous cycle on the community and we really want to push this forward because I think that could really have a very nice potential for everybody involved.

**Craig:** 27:35

Yeah, I mean it's exciting to watch. It's the most exciting thing I've seen in robotic control to date. Is it open source, RT2 or how can people work on this?

**Vincent:** 27:52

Our T1 is open source. This is a model that's smaller and more portable and that we've just open sourced the training code for it and we have also checkpoints and models for it so the community can leverage it and make derivatives of it and play with it. And we're going to continue sort of open sourcing a lot of those models, Because I feel like we're in the very early days of this revolution and we really want all the community to build up from those models.

**Craig:** 28:35

To Boston Dynamics who, at least from the public view, has the most advanced hardware. Are you able to integrate this with hardware systems like that that are much more versatile?

**Vincent:** 28:58

So Boston Dynamics is not part of Google anymore, so we don't have access to the robots. They're a fantastic team and they are very impressive robots. We have a number of robots that we work with To some extent. We try to work with a very diverse set of robots because we have this hypothesis that the diversity of experiences and of embodiments is really the key to unlocking the potential of those kinds of models. So we try to be very hardware and nostic to some extent and bring in a lot of partners into the mix and a lot of various diverse form factors. We're really focused on the brain aspects and bringing the intelligence to bear.

**Craig:** 29:50

OK, and last question: what's the next? Are we going to see another announcement in three to six months? Where are you going with this?

**Vincent:** 30:00

The entire field is very green. There are a lot of directions to explore and, you see, what I think you can expect is that the progress is going to mirror the progress you see in large multimodal models like Gemini, and those are evolving rapidly, and the capabilities that are enabled by those models are evolving rapidly. What we've shown with RT2 is that robotics can track with those improvements, because we're basically a thin layer on top of the advances that those models provide, and so we're going to expect a lot more in that direction in the future.

**Craig:** 30:44

OK, Vincent, I really appreciate the time. It's one of the most fascinating areas I'm reading about. If we could have another call at some point I would love it, but I'll let you go now. Ai might be the most important new computer technology ever. It's storming every industry and literally billions of dollars are being invested. So buckle up. The problem is that AI needs a lot of speed and processing power. So how do you compete without cost spiralling out of control? It's time to upgrade to the next generation of the cloud Oracle Cloud Infrastructure, or OCI. Oci is a single platform for your infrastructure, database, application development and AI needs. Oci has four to eight times the bandwidth of other clouds, and offers one consistent price instead of variable regional pricing. And, of course, nobody does data better than Oracle. So now you can train your AI models at twice the speed and less than half the cost of other clouds. If you want to do more and spend less, like Uber 8x8, databricks, mosaic take a free test drive of OCI at oracle.com slash. Ionai that's E-Y-E-O-N-A-I all run together. That's it for today's episode. If you want to read a transcript of the conversation, you can find one, as always, on our website IONAI that's E-Y-E-O-NAI. In the meantime, remember, the singularity may not be near, but AI is changing your world, so pay attention.