Google's Open AI Strategy — Omar Sanseviero, Google DeepMind

29 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=HXLAYQS-IAS
https://www.youtube.com/watch?v=Hxlayqs-IAs

SUMMARY

Gemma 4 has been released as the most capable open model yet, featuring advanced multimodal capabilities and an innovative architecture that allows for efficient parameter management. The model can run on various devices, including smartphones, and supports a wide range of languages, enhancing its accessibility and usability.

- *Gemma 4 introduces effective parameters, allowing for quicker inference by offloading some parameters to CPU or disk.*
- *The model is designed for on-device applications, optimizing performance for smartphones and similar devices.*
- *It supports multimodal inputs, including audio, images, and short videos, with improvements in object detection and speech recognition.*
- *The integration with Android Studio enables developers to use Gemma 4 for coding assistance directly within the IDE.*
- *The model's tokenizer is highly effective for multilingual applications, yielding better results for specific languages compared to other models.*
- *Future developments may lead to more powerful models running directly on devices, enhancing user experiences.*
- *The team emphasizes collaboration with various partners and community feedback to improve model capabilities and integration.*
- *The conversation touches on the evolution of fine-tuning practices, suggesting a shift towards using models out of the box rather than extensive customization.*

[music] >> We got so much Gemma 4, Gemma 3 1, Gemma scope med Gemma. Give us the TLDR. Yeah, so yeah, Gemma 4 is just out. This is the most capable open model we've released so far. We really tried to compact as much intelligence per parameter as we could. Bring all of these multimodal capabilities. So yeah, that's Gemma 4. So one interesting thing, you have this thing with effective parameters, not active parameters. Can you explain what it is? Yeah, so pretty much in the traditional transformer architecture you have like this big embedding layer, right?

And this new architecture is is more of a small change in the transformer architecture, in the transformer block. Pretty much we add a per layer embedding. So at every layer we add an embedding table. What is exciting is that you don't need to do like the full matrix multiplication. This is pretty much a lookup table. So the Gemma 4 model is a E2B. That means that it effectively has 2 billion parameters loaded into the GPU. It actually has almost 5 billion parameters, but those 3 billion parameters can be in the CPU, they can be in the disk, which means that you can do inference extremely quickly. This is just a lookup table. And what's the con?

Why don't we Why don't we always do this? Can it scale? Is it open research? Like you know, it seems very Okay, if I can just offload half the parameters to CPUs. Yeah, so pretty much here we did lots of quality experimentation and this is really optimized and designed for like on device. And when I say on device I mean like running in a phone, Android, Raspberry Pi, and so on, right? When you go larger you usually want to compact more You want to have more like dense architectures or MOEs.

So this this research This research decisions were very helpful for these small small use cases. Yeah, something I learned from the run that you organized this morning. For for our listeners, I think it's the first ever like official run club at AIE 6:30 a.m. Very rough, but at least I woke up for it. I met Cormac and he was telling me that I apparently in China the super apps are shipping models in the app bundle. For inference and just like use among all their super app. Assistants.

Yeah. And I don't know is is is that like a target use case for you guys? Yeah, so actually if you install like if you buy a pixel phone or a high end Samsung, they come from with a Gemini Nano and Gemini Nano is baked into the operating system and Gemini Nano is really built on top of Gemma. So last year we released Gemma 3N which was this architecture really designed for phone use cases and they use a Gemma 3N with some additional training, some additional adaptations to make the model good for like traditional on device use cases, right?

So pretty much when you buy like these high end phones, you can already use a Gemini out of the box. Yeah, we actually covered the 3N paper in our paper club and this like idea of like sort of parameter offloading or like download on demand is like very cool. Is it exactly the same in the Gemma 4 stuff? Yep. Okay. For the smaller models. >> Yeah. Yeah. Yeah. And does it does it scale? Is there a potential So for reference, Gemma 4 is a 29B and a 31B ones and only one's dense, but have you scaled it? Have you pushed it up? Is it We are doing lots of experiments.

>> Experiments. Yeah, yeah. Stay tuned. Yeah. What goes into shipping a mean line model like this? Like Yeah. What what's the behind the scenes? It's complex. The Gemma team is actually relatively small. We have like two or three PMs, we have one marketing person and then there is our like engineers and researchers working on shipping this. Of course there's like the full training part, we how do we do the post training, distillation, post training techniques and so on.

What is quite exciting is that once we have the model, then we collaborate with a bunch of open source partners, right? So for example, we work with a Lama CPP, Olama, MLX, Hugging Face, vLLM, Nvidia, AMD. So we have almost 50 external partners for every well for the Gemma for lunch, which has been the most complex launch. And also internally, we collaborate with a bunch of different teams. So, think of Google Cloud, Vertex, Vertex models models as a service, ADK, uh and then Android as well, right? So, we work, for example, with Android team and uh with the launch of Gemma 4, we released an integration with Android Studio. So, in Android Studio, there is this agent mode where you can have a a model helping you write code and do things within Android Studio. And they ship this integration with offline models using llama.cpp or vLLM or any open AI compatible endpoint.

So, now you can use Gemma 4 to also write code Android applications in Android Studio. What's the difference? When would someone want to do that versus just using Gemini? Outside of course Outside of the obvious, you're offline or you want the privacy. >> planes a lot or something. I did. Okay, I will say, on my long 10-hour flight to London, I did use Gemini as >> Yeah, I I was on Gemma 4 though. Sorry, Gemma Gemma. Yeah, yeah, it's mostly offline use cases. Right or if you Yeah.

Offline or privacy, like if you want to have all of your development set up locally and you don't want to send any code to to any API, you would use that. Do you see a future where, you know, small models get good enough? Like, does it cannibalize? It's an interesting position. Like, you have big Gemini, you have Gemma, both get exponentially better over time. Like, current Gemma is much better than what we had closed source a few years ago, right? Yeah, for me, it's quite exciting. I mean, if you look at Gemma, you compare to how we were 1 year ago, I would say Gemma uh 4 is matching state-of-the-art from 1 1 and 1/2 years ago for most things. With local models or models that you can run in your own hardware, you can get capabilities, so you can get agentic agentic capabilities, function calling, system instructions, like conversational and that kind of stuff.

Knowledge is much trickier, so for knowledge, you do need a larger model, right? That's why if you compare Gemini to Gemma, Gemini It much better knowledge and of the world, right? Like facts, information, and so on. So, it really depends. I do think we are heading towards a future in 1 to 2 years where Imagine like you can run a Gemini 3 Pro powerful model directly in your phone, right? And I think once we get there, things will be quite exciting.

From a product integration, from which experiences we can enable to users. I wouldn't say it cannibalizes. Still like two very different things. Like if you want like flagship capabilities, this super complex, long-running task you would do with Gemini if you need factuality and so on. But I do think for many of these agentic things, we'll get to a point in which we can do very powerful things directly on device. Yeah. Can we talk about the multi-modality side? Any advances there that you want to highlight or you've been getting good feedback on? Yeah, so Gemma 4 was built on the same research as Gemini 3, which pretty much means that we benefited from all the improvements that happened with Gemini 3. Multimodal wise, the smaller models can understand audio, images, and short videos. So, 30- to 60-second videos and and audios. For us actually quite long.

And even the on-device, like the 2B, 4B, 2B can do very good multimodal. Yeah, for audio we have a speech recognition, we have a speech-to-translated text, and then a bit of a speech understanding. So, you can do like a ask questions about an audio file and so on. So, use cases that are very optimized for like on-device phone use cases. And then on the vision side, we also improved things quite a bit. So, we have object detection, pointing, captioning.

We do not have image segmentation, which we know is like one thing that many people have been asking us. But otherwise, like for many things, we do support that. The other thing we do not support yet is video with audio. So, we can understand like video input or audio input separately, but if you want to pass like in the same prompt both a visual part and the audio part, we still need to do some improvements around that. And that's just a matter of like more data?

Probably some additional fine-tuning could yield some very good baseline models for this. Yeah. Yeah. What about audio out? We are exploring some things here. Nothing I can share at the moment. Yeah. >> [laughter] >> I think everyone's excited about the like when do you have native speech-to-speech, right? >> Yeah. But as far as I see, people always get excited and then the pipelines always win. Yeah. >> [laughter] >> Yeah. Yeah. Gemma is quite important for the multilingual aspect as well. So Gemma supports these 140 languages. Uh You did a lot of work on the multilingual order the tokenizer. The tokenizer, right. Right. So adding Yeah, exactly.

So the tokenizer has been pretty much based on the Gemini tokenizer. It's extremely good. So independently of the Gemma capabilities, if you just pick base Gemma model and you fine-tune for an additional language, it actually works extremely well. What are some So I didn't read that part. What what are some insights on the tokenization? And this comes from Gemma 3. Like this has been done already for over a year, but the tokenizer is pretty much the same as as Gemini, which means that the tokenizer lends itself to capture the right tokens for different languages. It's like a very good multilingual tokenizer. Which means that if you compare Gemma 3, so I'm going to the previous generation. If you compare Gemma 3 to other models from back then, maybe the other models were better than Gemma 3 like as general model, but if you train all of these models for, I don't know, a specific Southeast Asian language, I don't know, Vietnamese, let's say, Gemma would yield better results even if the other base models were potentially better. Yeah. I mean, I I I think there's some limit at which you basically have tonic representation, right? Like you understand the core concept and it translates to whatever language you want. Yes.

>> [laughter] >> I guess, you know, you are also you have purview over all of uh Google developer experience. And you brought the team here for the first time. What was that like? It's quite exciting to be honest. We have already participated in previous AI in your conference like Philip or like other team members have been in some of these in the past. This is London. This is DeepMind's home. You have to. Yeah, we have to. I mean we brought a bunch of researchers from the team to share about different things that we're working on. We brought other teams from Google not just from DeepMind that are also like using AI in one way or another. So we brought people that are doing on device machine learning, people that are doing like artist or optimizations to run models directly in phones or in the browser. We brought people from the Android team. We brought people that are working all over Google from robotics to research to Android. So yeah, it's quite exciting to come here and really show all of the things that the the company's building not just come and share like the things that our team is doing but really all of the overarching AI story that we're >> Yeah. I think you are I mean it is the lab with the biggest scope. Like you do everything including dolphins.

And it's very impressive. Like yeah, so you brought Sander. We talk a little bit about the researchers that you brought. We brought researchers in a couple of different topics. So we brought one of the researchers that worked in the Gemma development in the development of Gemma 4. We brought a researcher that works in the fusion models as well diffusion transformer models or diffusion as text generation not not image generation. >> Which was announced but not released.

Exactly. We did the Gemini diffusion last year at the IO which is very cool because you can generate code extremely quickly, right? Like it's a yeah, stupid. >> So the main pitch is speed but other than speed is there like

a secondary, you know, what can we do with a diffusion model that we cannot do with auto aggressive, you know? >> It's mostly speed. Okay. Yeah. I I feel like in terms of code structure there may be some things where you're like, okay, I want the brackets here. Yeah.

And then you fill in the blanks, right? So fill in the middle is like a common code problem but this is extended fill in the middle, or like extended like oh, help me upscale or put a Laura. I don't know, you know, translate the image analogy to text. I think in the past fill in the middle was like this task that many companies were trying to tackle as an additional generation task. And now people are just assuming that the model can do fill in the middle with a general exactly. Yeah, yeah. No no tricks about special tokenization or anything like that. Exactly. It used to be a, you know, mass language model that you you're trained to predict fill in the middle. Yeah. You had to rearrange your data set as well in order to to do FIM.

Yeah, it was a bit tricky. People were always getting like the the prompt in the the the the tokens wrong and yeah. If you deviated in any way from the training format, it didn't yield good results. Now we have like very good out of the box capabilities for that. What's the What's the idea about investing in text diffusion? Is there a world in which this overtakes auto-regressive? Yeah, yeah, that's a good question. I think at the moment it's still very experimental. I think we'll be really seeing us sharing a bit more research of the things that we have been doing around diffusion generated text generation models on that space. I would say it's still very early stage. I think especially like the model quality is still a bit worse from what you would get from the a normal auto-regressive model. Yeah, a lot of what you were mentioning earlier about it for, you know, okay, fill in this code block, this stuff. It seems different how we're building agents these days of, you know, sequential tool calling, this that.

I guess it's if it's just speed, it's speed. If it's an honest I could see I could see a world where there's like system one, system two. System one is the diffusion one, system two is auto-regressive. System one is the the planner, system two is the executor. I don't know. Yeah, could be. Maybe it's it's it's too hypothetical at this point, right? You know. But diffusion diffusion transformer models are difficult to fine-tune as well. So so there's also like a point in which how much flexibility Yeah, I could see a world in which you just have like a very strong agent manager kind of set up and then you have like executors like diffusion based executors that do like a specific coding. Are people fine-tuning outside of you know we see a few big companies do okay like cursor has a really good consistent or there's a few that have done fine-tuning but it seems like it's not picking up as you know Yeah so there was this period 2024 I think in which there was like this maybe 2023 like there were all of these fine-tuning communities and I think it's been changing quite a bit over the last two years because models are getting very good out of the box so as I was saying like for Gemma 4 we had 50 to 60 partners and some of them were like oh yeah we're going to try and fine-tune the 27B model for this vision task and then they were like oh actually the model works too well out of the box we don't need to fine-tune it. Yeah we saw lots of those things so I'm seeing less excitement around fine-tuning nowadays as general conversational models. Yeah. There is still quite a bit of excitement around fine-tuning for specific domains like finance health care specific types of data that the model didn't see but as general conversational like just changing how the model behaves you can do most most of that via prompting nowadays and in terms of capabilities the models are very good out of the box so it's been changing quite a bit there's still like the onslaught people I don't know if you know Daniel Khan and Astro Michael >> workshop to just talk about >> Yeah yeah they they are the goats they still do like amazing

tools for the community to fine-tune and the community use those tools but I'm seeing like some changes in the trends I think people are not fine-tuning that much anymore. And you guys put out a version of your own Med Gemma is a fine-tune of Yeah yeah so Med Med Gemma the last Med Gemma which we released three months ago Med Gemma 1.5 it's based on Gemma 3. Gemma 3. Yeah Gemma 3 so it's pretty much Gemma 3 and then additional training with some of our medical data sets. How do you see if I'm not mistaken Apple Foundation models on device were a bunch of Laura's for different tasks and when you're constrained or running on device small efficient models you guys did a offload so you're like caring about efficiency but you know do you see a world of multi Laura's for tasks should people be fine tuning the small one?

I think this is a big challenge in general in the whole developer ecosystem because let's say that you want to have 20 apps in your phone right and let's say that each of those apps comes with its own Laura right? What happens when you update the model base model? You also need to update all of these Laura's so from a developer point of view I think it will be very tricky because when you don't want to have 20 different base models in the phone of the users the battery will just die you also don't want to have to update 20 Laura's every time you update the base model right? So the release cycles in the Android world are and in the iOS world are very different so yeah I think it's more of a general industry challenge that we need to figure out how we think that people should build ML like on device phone power like experiences.

Yeah it's it's more of a product and developer experience kind of challenge. Yeah I have a question about the bigger Gemma models so you have two models that are pretty similar size one is dense one is a MOE can you talk a bit about okay say you have a 27 B you're putting out how do you think about should I build an MOE outside of inference and using it how do you think about when to do MOE versus dense what are the trade-offs? What else is inference what else is there?

But then there's two at the same size No yeah but I mean one is 31 B which is dense right? And that's like the most raw intelligence and then you have the 27 B with a 4 billion activated parameters But like you know why not a 31 B 5 B active for example? Yeah I mean you you can just fit more in dense Yeah I mean we did quite a bit of experimentation and like research and like which would be the best sizes that would be friendly to developers and which also we we made decisions along that right?

The 31 B is really like the largest model size that quantize would fit in a consumer GPU. The 27B is more like an extremely fast inference within those constraints. MOEs are challenging to fine-tune. I don't know if we've talked about that in the past, but MOEs in general are like an extremely good AI architecture. They work great for inference, but when people fine-tune them, they struggle a bit. Like they are not as easy to fine-tune for instruction following. The standard recipes and hyperparameters that you have may not work out of the box for MOEs. The intuition is the the routing kills the backprop or I I think so.

I I don't have a very strong intuition on it either, to be honest. People always say this, but I'm trying to say why, right? Because if you can train it, you can fine-tune it. Like Fine-tuning is just training. Yeah, I think I think it's a mix of the routing and yeah, just having like different distributions. The distribution may affect the routing in a different way than a dense model where you just change the things. Uh That's kind of my intuition, but also like

I think there are many different variables here. Like how many uh experts do you trigger or uh Yeah, there are like a bunch of different parameters that you can move.

Like whether you freeze or you don't freeze the the router, like a bunch of things that you need to to think about. Yeah. To me, the most important asymptotes that I'm looking for are what is the minimum sparsity level that we can reach, and then what is the most let's call it ELO per byte. Yeah. Yeah, yeah, that's the thing I mean plays quite a bit. Like what's the intelligence per parameter, right? Like how do we maximize this intelligence per parameter because Then we can stop, right?

Yeah, [laughter] because if you compare like the I mean Gemma we have done the same size, right? 27 like almost 30 billion around 30 billion parameters for Gemma 2, 3, and 4. And the intelligence is much higher, right? Like we have not increased the model size. >> that that >> Yeah. Yeah. It was an easier number when everything was dense. Then you have to add in sparsity. Now Now have offloading. Yeah. So Yeah, you cannot compare like MOE students models.

There's there's no There are some like napkin calculations you can do to compare, but it's not apples to apples, but That's a good question. Like I don't know like where we'll be in 3 years from now. I would assume like a 30 B parameter model parameter model could be extremely powerful. I still think there are limitations in terms of knowledge. So maybe the model will be able to do it like >> Yeah. super wild gigantic stuff, but it will not know like who was the president of X country 25 I mean maybe yes, but like very niche knowledge probably the models will not have. Yeah. It's this is this is this is just information theory, right? Like you're using the model as a database. So of course there's going to be limits. The other thing is also I always think about when we talk about this topic superposition, right? And topic has this the concept of superposition where you can store information in the smaller weights as because it compounds with the other weights as well. Yeah. And so not that much research on it since then, but maybe this is my segue into Mechanterf. GemmaScope.

>> [laughter] >> Yeah, so last year in December we released GemmaScope. So GemmaScope pretty much allows you to analyze the the activations across different layers based on the tokens in those layers. And yeah, the team released I don't know if it was a couple of terabytes, maybe even up to like one petabyte of data that we had to store because we did that for every single layer across all of the Gemma 3 models. So it's a very complete >> And Llama as well? We did it just for Gemma 3. Oh, okay. I I think Neuropedia had some others.

>> Could be. Could be. There's a little teams I was like wow that was a yeah, it was just a very very cool like cross-lab partnership. Yeah, yeah. There are a couple of open source tools as well that you can just do create your own yeah, your own activation networks. Yeah. Yeah, yeah. It's a niche field. I think it's a it's a good opportunity. I think we were talking about this earlier, right? Like it's an area where you don't need lots of compute to get started. That allows you to understand like how of model works. You can experiment. You can get a bit of a sense of how we have all transformer architectures works. So it's a good area. Okay, the context of this is really like why bring researchers to AI engineer, which is an engineering applied AI conference.

One to me, it is actually very important that you bring the researchers because engineers want to learn about

how the models would that they use were trained even if they never ever trained it themselves. >> Mhm. Right? Because I think they just feel more trusting of the model Yep. if they if you peel back the curtains a little bit. And also I think there's some prestige that people want to feel like they can go home and talk about it intelligently even if they if they don't actually, you know, not know how to train it. The other thing is like I do think that research and engineering are closer than people think. Yep. Uh there's I mean there's research engineers. Yep. And Mechatronics is probably the easiest single easiest way that engineers can get into research if they want to. Yeah, I think I mean in in big part like so many researchers are doing ablations, right? Like they are just moving the pieces around and seeing what works and what doesn't work.

Of course there's like a branch within research that is more much more like architecture design and like a much deeper. But there's lots of very like empirical experimentation and seeing what works, what doesn't work, moving things around, which for me is much more engineering rather than for like research unless we are like writing new activation functions maybe, but >> Yeah, I think this maybe is a change in your career as well. Like it used to be a like a joke like haha researchers are terrible at coding and then they throw it across the wall to some engineer that will that will clean up the code.

>> [laughter] >> Yeah. But now everyone has their own personal research engineer, right? >> Yeah, and something that is cool that is happening is also how researchers begin to adopt some of the cool agentic tools now. So for example, within the team we are building a skills to do experiments and ablations and evaluations and how the research team can use all of these agentic tools as part of their research process is also quite interesting. >> Yeah, yeah. I had ETA on my podcast who led the post writing for IMO the IMO gold model. I think it was DeepMind.

And he was notably he was an AI researcher that doesn't use AI. Yeah. It's It's gone even further. People making novel math research like some of the Erdos problems. They're engineers not researchers with no background in math. Just you know using coding agents. among the math guys. Yeah, just not [laughter] math not research but you know solving some of the most you know Unsolved problems. >> Unsolved problems in But even in the model architecture side of things like 2 years ago when all of these people started to fine-tune models and to do experiments and do model merging there was quite a bit of research that was happening in GitHub and in Reddit and in local lab and people were actually like inventing new things and then there were papers Frankenmerges Yeah, like all of the franken merge stuff like all of the axolotl library like all of these tools and there were papers published by different companies and research labs 1 or 2 years later that were rediscovering what was already done by the Reddit or Discord people without yeah anyone noticing.

Do you have a take on auto research? Every AI wave has an auto ML wave and this is the auto ML wave of this wave. Always been a bit skip I mean auto ML few years ago was mostly like just a parameter search search yeah yeah pretty much like ready search in the in this higher yeah parameter space uh I don't know like with Karpathy experiments it's been quite interesting to see like how things are evolving. I will I don't know what's your take on this. Things are just cooler when he does it.

I do think some some part of this is you're just speed running experiments agentically right? The agent The

coding agent is more autonomous you can actually go to sleep and it will do the things that you would have done anyway. So you're just kind of automating things that you would have done. I see it differently. I think okay it will be a very exciting time if we have a move 37 from an auto research. If you make an impactful discovery that someone wouldn't have thought of right? So there's a side of okay I have these ideas go run them in the background that's fine. But the the The side is actually when you're shooting off not just paths that you wouldn't have thought about, but you know, trajectories that people wouldn't think about and they work and you make new discoveries. That's the very exciting thing. I think when you have more approach to just token spend and send off, you know, hopefully that becomes possible. Yeah, I do think the next generation of fine-tuners will not be like I mean will be people that are not coding at all, right? Like 1 year ago we had to write like our own code that with transformers or on slots or or whichever library of your choice.

I do think as we like keep evolving like most people will be fine-tuning with a couple skills, right? Like Hugging Face has a skills, like all of these libraries have skills. They will just uh prompt the agent to kick off like some experiments and see what works, what doesn't work. And honestly, it's a it's a good middle ground right now. Like all the tools you've mentioned, they let you fine-tune in minutes. You don't need to know what's happening in the order. Yeah, so I think that's where like the direction will be like people that just want to do fine-tunes to improve their capabilities for certain domain or like add some like new behavior. They will not be coding the fine-tuning code, but of course if you want to do like deeper research in the architecture, my hunch is that most likely uh this will not be like automatable at least in the next 1 or 2 years. Okay, we got to wrap up soon. Uh I just wanted to end a little bit on your your your growth in your team. Um and you know, Page is here, uh Logan is is over in SF, uh and you've been hiring all my friends. Uh Thor and Ivan and all these um what does the team look like?

Where are you looking to grow? It's been quite exciting. We are hiring lots of very high agency people. Yeah, I think maybe 3 4 years ago we we did have like a nice interview about how I was growing like DevRel at Hugging Face and how we were thinking like DevRel should look like. DevRel at Hugging Face is also interesting. It's redefining what DevRel should be in an AI very AI-centric organization at the frontier. It's a research lab at the end of the day and we are in this AI era. So it's also rethinking what DevRel world should look like here in 2026. We are Yeah, we're having pretty much like high agency people excited to to to build things, to engage with the community, and so on. Right now, we're growing in Singapore, so we're looking to hire someone in Singapore, and >> Come to the AI in Singapore. They're coming. Yeah, yeah, and to hire someone in India. So, those are like two locations with huge >> Singapore so important?

So, Singapore is interesting. Singapore has a relatively small, but very high like very dense high-talent community. Now, we have a proper DeepMind office as well, like it's small, but it's growing quite quickly as well. Mostly because he doesn't want to move. >> [laughter] >> Yeah, yeah. >> But, there's a huge win for Singapore. We don't have research in Singapore usually. Yeah, so we're trying to grow the team in places collocated to like people doing like traditional DeepMind research activities. We don't want like to have like Sales office.

>> people that is in a single town that is not connected to anyone in person from from DeepMind. So, ideally, if

they go to the office, they can talk with researchers doing like their own thing. If it benefits a different project, they can be part of like the more DeepMind-y side of things. So, So, yeah. So, we have like people in in Paris, in London, I mean, Zurich, in SF, New York. So, all of these DeepMind hubs and Singapore now is becoming like a very small, but very exciting as well. So, this is all the DevRel, DevEx team in DeepMind, right?

>> Yeah. DeepMind has also expanded a lot here, like a few weeks ago, Kaggle joined DeepMind. How's How's the org in general, actually? DeepMind in the past didn't do that much product. And yeah, now we have like AI Even though the Gemini API. Now, Kaggle but Kaggle is is part of the team. Actually, Kaggle is also here. There are two team members here talking about the Very excited. Yeah, talking about evaluations. Uh, they last week they released a a new system for agent evaluation. It's like a very like experimental initial benchmark, but pretty much allowing agents to take an exam and compete in a leaderboard, which is always fun. Yeah, when with with Kaggle joining us, I think there are a couple a lot things.

There's a whole Kaggle community hackathon things that uh enables the community to build hackathons, but there is also the Kaggle benchmarks. And I think Kaggle benchmarks can connect very well with how we think about Gemini and the capabilities. And if you're in the eval space, then you know like many benchmarks can be benchmarks. Uh many people are gaming the benchmarks, and we want to identify like which are these capabilities that maybe we are not aware that we have or that maybe we could improve and bring all of that feedback from the benchmarks created by the community in an organic way and bring that all of that feedback back to to the model itself. Yeah. I mean the way we are doing Gemma, Gemini, and all of our tools is really like based on the feedback from the startups, the community, the developers. So that's why you see like Logan page, everyone in the team talking with the community in social media, in forums, in events, and really understanding what people are building uh with our tools and bringing all of that feedback to to the modeling things, uh which is very cool as being part of the team. Yeah. Yeah. Well, you guys are doing amazing work. Thank you so much for joining us here. And Yeah, thank you. Can't wait to see what's next.

Yeah, thank you for having us here. >> Yeah. >> [music]