/mlst

Panel discussion on ARC Prize 2024 (Zurich)

March 23, 2026



Machine Learning Street Talk

Podcast Transcript

Revision: 6e6e6807

Topic headings include retention indicators: • above average, • average, • below average (vs. similar YouTube videos)

Interested in licensing ReScript for your podcast? Get in touch with Tim Scarfe.

Contents

Overview 2

ARC Prize 2024 and o3 Performance 3

The ARChitects Winning Solution: DFS on LLM Token Space 5

o1/o3 Architecture Speculation and Inference-Time Compute 9

Reasoning in Continuous Space vs. Chain of Thought 14

Benchmarks, AGI Definitions, and Moving Goalposts 20

References 27

Overview

Filmed at Tufa AI Labs in Zurich in early January 2025, this panel brings together Tim Scarfe with the actual winners of the 2024 ARC Prize – Daniel Franzen and Jan Disselhoff (the "ARChitects") – alongside IBM Research's Michael Hersche, with Tufa Labs founder Benjamin Crouzier moderating.

The conversation opens with Tim explaining why o3's performance on ARC forced him to fundamentally update his views. He'd always believed solution-space prediction was impossible for ARC-style problems, assuming you'd need programs with compositional generalization. o3 changed that, though questions about dataset contamination and the sheer compute cost ($17 per task) remain.

The heart of the episode is hearing directly from Daniel and Jan about how they actually won. Their approach is elegant: they tokenize ARC grids line by line, then run a depth-first search through the LLM's token probability space. Because ARC grids have far fewer valid completions than natural language, there's a tight alignment between completion probability and correctness. They exploit this by searching below a probability threshold, augmenting problems (flipping, rotating), and multiplying probabilities across perspectives to select the right answer. It's the same model doing generation and verification – just with shifted viewpoints.

The panel then gets into informed speculation about what o3 is actually doing under the hood. Is it tree of thought? A single model doing self-play search? The consensus leans toward something like STaR (the Noah Goodman Self-Taught Reasoner approach) at training time, with sophisticated tree search during inference to find optimal chain-of-thought prefixes.

Daniel drops what he calls his "spiciest take": that chain-of-thought reasoning in discrete tokens is fundamentally the wrong approach. He points to Meta's Large Concept Model as more promising – thinking in continuous concept space rather than being forced to externalize thoughts as words. Michael Hersche pushes back, noting you're still missing proper state representation either way.

The discussion closes with the perennial question of whether benchmarks test the right things, what AGI even means (Chollet's "skill acquisition efficiency" definition comes up), and whether we'd recognize AGI if it showed up wearing a jagged intelligence profile. Tim notes that using ChatGPT Pro with o1 has been "genuinely unreal" – qualitatively different from anything before.

ARC Prize 2024 and o3 Performance

Tim Scarfe 00:00:00

I always said that if the ARC challenge got beaten, that I would update massively. And it has been beaten and I have updated massively. With some reservations, I'm still slightly concerned that there could be some dataset contamination. We don't know exactly what they did to fine tune on ARC, so there are some questions that remain unanswered. It's quite a classic thing for OpenAI to announce incredible things and then just like with Sora, 9 months later, you get the, you know, the shocking reality unveils itself. But with that said, it feels different. It feels like it's an example of true developer aware generalization because you can see that they are basically 1 shotting it transductively. So in solution space, you can see their prompt template. They just give it a few candidates' specifications and it can apparently produce remarkable performance, albeit with an insane amount of computation. So I guess we'll we'll wait and see. But what do you guys think?[1][2][3]

Jan Disselhoff 00:01:02

Yeah. I think it's quite impressive that they managed to do that without any kind of test time retraining like we do. So that the model can do it in 1 shot, I think it's it's very amazing.[4]

Daniel Franzen 00:01:13

I think it's also crazy. Like we are like all ARC people, so we are focused on the ARC dataset, but they also have crazy performance on different data benchmarks. And the 1 that impressed me most is the Frontier Math benchmark, where they get 20 as far as I know. And I I 25%. I I get 0. So it's a little bit annoying. It it will be really interesting to see it when it's get gets published. And 2 years down the line, when probably if the trend holds, the open source models get the same performance, it's going to be scary. Yeah.[5]

[1] ARC Prize
[2] Tufa AI Labs
[3] Jan Disselhoff
[4] Daniel Franzen
[5] Michael Hersche - IBM Research

Michael Hersche 00:01:51

I do not see yet is how it kind of like, it tackles reasoning well, apparently, But some of the kind of inherent weaknesses of items are still there, right? Let's say hallucination or so. I don't see yet how kind of an o3 addresses these issues yet. It could

Tim Scarfe 00:02:12

be like with self correction or so, but I don't know how would they really do it. But probably nobody knows so far. We now seem to have second order hallucination. So, noticed 1 Pro, it hallucinates from the reasoning trajectories, which is and it's even more insidious.

Michael Hersche 00:02:32

Yeah. And I think that's very hard to detect. 1 could do it if I met a model, but, yeah, I don't know.

Benjamin Crouzier 00:02:39

So you previously said that initially you were very impressed with the theory and then not quite as much. What caused that adjustment?

Daniel Franzen 00:02:47

That they used the training data of the ARC to fine tune. Like I thought it was like really true 0 shot training, and then I was on the floor. And now it's clear that they use a little bit of the training data at least, and it's slightly less impressive. And they also use a lot of compute and also slightly less it's still crazy. Don't don't misunderstand me, but it's it's not out of this world crazy. Like, I can imagine that you can get there from a current perspective.

Tim Scarfe 00:03:19

But it wasn't unfair because Gary Marcus went off on 1 saying that it was scandalous that they trained on the training set. And of course, it it it's completely legitimate. What what interests me is that Chollet has made a career out of being the critic. And even he has not fallen on his sword, but he is not he is not his usual self. He he does genuinely believe that this is a major breakthrough in generalization, the extent of which, of course, we'll find out.

Benjamin Crouzier 00:03:48

How did it look like, your process of updating? Because, initially, you you were on the, with, thinking that, like, need programs. This is program induction.

Tim Scarfe 00:03:59

So, Chollet's position is ambiguous because he still thinks that this is not an LLM. As in, it's an LLM, but it's some kind of a search controller. So as far as he's concerned, this has proven him right. If it turns out to be a simple, you know, autoregressive single forward pass of an LLM, I think that wouldn't look good for him. But the main main thing for me though is I always thought it was impossible to do solutions based prediction. I just didn't think it was possible. So I always thought we need to have a program because programs have all of these features that we know and love. Compositional generalization and that, you know, they run on Turing machines and so on. I never thought that these things could actually make strong generalizable predictions in solution space. And when you look at the failure modes on o3, if anything it seems to prove that they do understand. In some of the failure cases, they actually made more correct predictions

The ARChitects Winning Solution: DFS on LLM Token Space

Tim Scarfe 00:05:00

than the true labels. In other cases, they made slightly kind of, you know, off by 1 type errors. And also they degraded when the solution space grew too large, which seems like something that can be fixed over time. So if anything, the evidence is pointing to them working very well at reasoning. What do you think?

Daniel Franzen 00:05:20

Well, like you said that the the LLM the LLM gets extended in a way in o 3. Right? There's a there's some some type of algorithm that goes around it that produces the chain of thought sampling. And it's something we have also seen in our approach. Right? We have a LLM model at the core of our approach, but a lot of our score increases came from using external tools to extract information from the LLM better. Like, the LLM Smart can give us a very good probability for the next token, but on its own, it's not enough because we don't have good sampling technology at the moment. So we had to add a lot of complex parts afterwards to get it to produce the correct solution. And I think you can substantially increase performance in LLMs a lot of times by just having a smarter algorithm behind it

that checks the results in a way.

Tim Scarfe 00:06:21

Yeah, that's really interesting that you said that they had the capacity to understand these problems, but just It's quite interesting, isn't it, that you can ask an LLM to discriminate. Generating is more difficult. But even more difficult than that is traversing the token space between pockets of good things. So what these guys did in their winning solution was they essentially had to, rather than using this stochastic sampling, you know, where every single token you just do this minimum entropy search essentially. But you design this depth first search where you had to kind of traverse many, many bad steps in order to get to a good step. And But when you actually found that part in the token space, you could actually kind of ask the language model, do you think that this is the correct, you know, program? And and then it would give you the correct answer. So the problem was traversing there in the first place.

Jan Disselhoff 00:07:21

Yeah. In some way it was. So sampling is an approach that essentially did work too on ARC, but it wasn't that efficient. So if we have a correct solution that we need to find which has only 10% probability to be predicted during sampling, then we had to sample quite often to find it with some chance. And with the depth first search, it's essentially a way to search through all possible candidates with sampling probability above some threshold and to do it very efficiently. The LLM just predicts token by token and we can essentially do a tree search on that prediction. We predict the first token, we get a probability, and then we go further. And on our path, we sum up the probabilities, the log probabilities of all the tokens as a measure of the probability of sampling this completion. And that's actually where we cut off at some point. If fall below the threshold, we say this is not promising, we go back and we search for other solutions. And this way, can essentially find a lot of good candidates. And we hope that the good the correct solution is within them with this process.

Daniel Franzen 00:08:37

It's a very problem specific solution because in in language tasks, the probability of any completion is quite low just because there are so many possible completions.

Jan Disselhoff 00:08:47

Many completions that are essentially all valuable, well good for language.

Daniel Franzen 00:08:52

But in our case, we only have 1 correct completion. Yeah.

Tim Scarfe 00:08:55

Let's make sure

Daniel Franzen 00:08:56

that To

Michael Hersche 00:08:56

be clear,

Benjamin Crouzier 00:08:57

your approach, there is a direct mapping between pixel and the grid and tokens.

Jan Disselhoff 00:09:04

Yeah. Exactly. Yeah. We just tokenize essentially line by line. But with the numbers from 1 to 9, have an end of line token and an end of sentence token for the end of the prediction. That's essentially the tokenization method we use.

Tim Scarfe 00:09:18

And and and you were saying that in natural language, it gets quite interesting to think about the reachability space, like, you know, in in language models. These guys at Toronto I interviewed, they they came up with this Roger Federer game, which is where you need it to say, greatest. So Roger Federer is the greatest. What prefix can I come up with to make it say greatest? And it turns out that if you have a large prefix budget, you can you can make it say greatest. In fact, you can steer the model to say anything you want. But in natural language, the token space, it has many more degrees of freedom. Right? Because language is kind of weird and gnarly. But on something like the ARC challenge, there are not many degrees of freedom and there are hardly any trajectories. And so the challenge is just finding that trajectory to get to the solution.

Daniel Franzen 00:10:05

The probability we talk about, like the completion probability, is just the sum of the logins of the tokens. Right? And in natural language, if you have you can have 10 sentences that all mean the same thing. So they can't have the correct probability if you think about the concept. Like, the concept has to has the sum of the 10 completions, but each completion is obviously lower. Right? In the ARC contest, because our our completions are so restricted in the token space, right, we have only 1 correct path, and then there's a large alignment between the probability of this specific completion and if the model thinks that it's the right completion. And we could leverage that to not only evaluate a completion from 1 perspective, but then also shift perspectives. Right? The augmentations evaluate it again, and then calculate the product of the probabilities basically to select our right candidate.

Tim Scarfe 00:11:03

I interviewed Randall Balestriero at NeurIPS, and he's got this interesting research that, you know, we think that we need to have these pre trained massive language models that, you know, know about everything on the internet. And if you're doing discriminative tasks, can start from scratch because the model is even from a blank slate. They have all these inductive priors and biases and so on. And they they just work out of the box. So on simple tasks, can just do a little bit of training on a Lama, you know, style model, obviously not pre trained. And you can do really, really well. And you guys found a similar thing. So I I think you suspect that even if you started from just a blank slate transformer, fine tuning it on the specific arc challenges actually moved the needle significantly more than any transfer from the base data.

Jan Disselhoff 00:11:49

We didn't try it.

Daniel Franzen 00:11:50

Not yet. We actually don't know.

Jan Disselhoff 00:11:51

It's just something we just everything and says the concepts come from the language that the model have learned, and they are just applied to ARC. But I don't know if

o1/o3 Architecture Speculation and Inference-Time Compute

Jan Disselhoff 00:12:00

everyone has ever tested if this is actually true or we just start with a blank slate model and train it on ARC. How well does it perform? Something we want to try out.

Audience Member 1 00:12:09

Yeah. But we haven't gotten to it yet here. Now that I think I understand your approach, would you mind explaining why you think you achieved 0% on frontier math? Because I think math is very similar where you have very sparse good candidates. Right? There's only so many formulas that are inherently correct versus natural language, there's a lot of combinations.

Daniel Franzen 00:12:30

No. I think that was a misunderstanding. I, as a person, can't get get 0% of what you're not. Yeah? I I'm stupid than a llama, so it's But

Jan Disselhoff 00:12:40

we are I still think it would be difficult for the model model because in math, there's still a lot of completions. There's still a lot of ways to write a formula. If you write a plus b or b plus a, that's exactly essentially the same problem as in language. So this approach would probably not work that well without some without lot of additional tuning.

Daniel Franzen 00:13:00

That's a research idea we are looking into, actually. Okay. Yeah.

Audience Member 2 00:13:04

So what's your current best guess for what 0 1, 0 3 look like architecturally? Like, are they doing at training time?

Daniel Franzen 00:13:11

I'm not sure. I I personally think it's just real life trained KPT4 model that has some very smart training data. It's a possibility, I think about. Like, we had a lot of discussion about it today, and there are different thoughts about it. But I see, for example, do you know the QWQ model from Alibaba? It's also a thinking model. It's okay. It's quite good for open source 32 b model. And I think they just train on tree of thought completions. Like, they do a tree of thought for a specific problem, they check if it's a good solution, and if yes, then they reinforce it. And it might be everything that o 1 is doing, or they have like a more complex back end. It's also possible that they do some very, very smart tree of thought in the background, which we never see. We don't know.

Benjamin Crouzier 00:14:11

How do you think that you do the verification? Is it like encoded or is it a model to verify whichever chain of thought is correct?

Daniel Franzen 00:14:19

I think the standard tree of thought approach uses a verifier, or the best tree of thought approaches I've seen. They use a verification model.

Benjamin Crouzier 00:14:27

Is that a separate model? Yeah. It

Daniel Franzen 00:14:30

has to be because the model is bad verifying its own outputs. Right? Yeah. That's also what we saw in ARC. We couldn't we couldn't just take the probability on its own because it would just be the probability. We had to shift the problem to model sees to get a good evaluation of if it's correct.

Audience Member 2 00:14:49

Or maybe it's just that poor people's models are bad at verifying the outputs. And when you're GPU rich, you can actually verify your outputs with the same model just fine.

Daniel Franzen 00:14:57

Yeah. Just how more data?

Michael Hersche 00:14:59

How different has the model to be?

Daniel Franzen 00:15:02

I I don't know. I have no idea.

Michael Hersche 00:15:04

I'm totally like yeah.

Benjamin Crouzier 00:15:06

So what does it look like in the case of your work on ARC? You said that you use a different

Daniel Franzen 00:15:10

No. No. In in our case, it's the same model, but we we changed the problem,

Jan Disselhoff 00:15:29

for some, the model will detect essentially that they were wrong and up to low probability. So if you multiply them, essentially get a good selection process there with using the same model.

Audience Member 2 00:15:42

So I guess as a follow-up on the inference and architecture question, the the spending according to their numbers, like thousands of dollars per arc task. I'm I'm sure that's an average, but, like, there's only so many ways to spend that much money on a single forward pass essentially. Right? So, like, that's a load of GPUs, a lot of power. So they need to be doing something. Right? They need it it needs to be doing something like, what are they doing? Like, it needs to be branching out a lot. And you can only have context that large. Like, what's your guess? Which way are they scaling? Are they just scaling all the ways?

Daniel Franzen 00:16:11

I think the cheap 1 was $6,700 for all all 400 Ag tasks.

Jan Disselhoff 00:16:16

17 per task on the evaluation.

Daniel Franzen 00:16:17

$17 per task.

Benjamin Crouzier 00:16:18

I think

Jan Disselhoff 00:16:19

$22,002 280,000 token or something like that. Yeah. Yeah. So That is quite a lot. And I don't necessarily think that wasn't 1 forward pass. Maybe they did.

Audience Member 2 00:16:31

Oh, yeah. For sure. Many many, but I'm just saying, that's what they're doing. Right? They're essentially doing forward passes somewhere. So you need to be doing, like, shit tons of forward passes, even if you have a lot of GPUs. That's

Daniel Franzen 00:16:41

Several books' worth of text that I produce. And if you have thought that long about the problem, then you better solve it. Yeah. But it might be at the moment, my my guess would be that they produce intermediate results, because each act challenge uses an annoying amount of tokens. Like, you have a 30 by 30 grid, you need 900 tokens at least to just print it. So maybe they just have like a thinking process that also produces intermediate results and then evaluates them, you know. And then just repeats until it thinks that it has found a good solution.

Jan Disselhoff 00:17:19

Could also imagine that the letter model reason, given the example that's that the model reason what what the actual task is, what the transformation rule is, and then ask the model itself to apply it to the data. So maybe something like this. And then iterative refinement or something like that. Yeah.

Audience Member 2 00:17:37

That would still be a general procedure, though. Right? I mean, we we know the prompt, so it's not in the prompt unless they're lying.

Daniel Franzen 00:17:44

I know. It could be very general, like, that you separate the problem into smaller steps, think about these steps, apply these steps, then just repeat until you get to a clear result. But nobody knows.

Audience Member 2 00:17:57

Well, some people know. They're just the same.

Tim Scarfe 00:18:00

Yeah. There are some hints on Twitter, think, Rune and a few OpenAI employees. And I misinterpreted their tweets. Just to be fair, by the way, I originally thought it was just a single forward pass because I I'm looking at the Google thinking model and, you know, QWQ and so on, and they they are a single forward pass. But they don't do much thinking. And I just know from experience that, you know, to get coherent anything for a long period of time is very, very difficult. And these things are thinking so much. What they said on Twitter was that it's all 1 model. And I believe that. I think it is. I think the model goes into a mode and they can do this tree search in that thinking mode. Because per Chollet, what's actually happening on the final unroll is you give it in the prompt a chain of thought program. So the search the search process is about finding what is the optimal program, which if I prefix this on my unroll, it will solve the problem for me. Comes back to that what's the magic word paper that was just talking about before, that there's almost like a secret key that if you prefix the language model with this magic sequence of tokens, it'll it'll do the thing you want it to do. So it's about finding that. We all agree at training time, it's doing something like STaR, the Noah Goodman paper. Right? So it's doing some kind of process supervision. It goes into a mode. They do this kind of, tree search. And I guess what they're doing is they're just trying to find the best possible chain of thought prefix. And then when they find the best prefix, given their computational budget, they stick it into the same model and they just unroll. And I think that's what they're doing.

Audience Member 2 00:19:30

So why does no 1 else do it that way? Like, why is Germany so Germany is so far behind?

Tim Scarfe 00:19:35

Well, that's interesting that they say the headline metrics are very good doing it the way they are doing it.

Audience Member 2 00:19:41

Well, they say many things. Yeah.

Tim Scarfe 00:19:43

But I think part of that is just having a model, you know, trained with this STaR approach to imagine the best possible train of thought, and then just immediately running on the first train of thought you came up with. Apparently, even that gives you a lot of uplift. And I'm guessing just

Reasoning in Continuous Space vs. Chain of Thought

Tim Scarfe 00:20:00

optimizing on that chain of thought program before you unroll is even better.

Audience Member 2 00:20:04

Can I ask a related question? Just what's your personal take? Who's gonna be the best reasoning model '25? And who's ready to put down money?

Daniel Franzen 00:20:15

I think it's a little bit of a I think the main difficulty of this question is just in which iteration we are at the moment. Like, it's always open my eyes, almost always at top, and then the other companies sometimes, like, spike above it, and 1 month later, OpenAI takes over again. So if we are in the 1 month spike where OpenAI loses, then not OpenAI, but most probably OpenAI.

Tim Scarfe 00:20:41

There's something magic about Sonnet. Something magic about it. I I can't quite put my finger on it. But it it just has the vibes that no other model has, you know, for a base model. Something magic happens when you do this thinking mode. The the models take on an entirely different character, but a bit of a soulless character, You know, they have some interesting properties like you know, now they're more creative because they're considered different options and so on. They're not as pointillistic, but they have no soul. Anthropic has this character to it that I've not seen anywhere else. And I'm very excited to see what happens when they drop their newer, better sonnet and with the thinking on the top.

Benjamin Crouzier 00:21:16

And of course, about DeepSeek as well. I expect them to do well, maybe not on the top, put out a good model.

Tim Scarfe 00:21:24

But now it's the context though that apparently DeepSeek, while it does well on headline metrics, it has no soul. Yeah. There's this guy called, you know the guy who wrote the Simulators paper, Janus? And he's a full time prompt whisperer. So he like, he just talks, he obviously thinks the models are conscious and all this kind of stuff, you know. But no, he he like understands these models in a way that no 1 else does. And and he just said that deep seek is like basically soulless.

Audience Member 2 00:21:51

I mean, that's science.

Benjamin Crouzier 00:21:55

A very common thing you observe if you sample a lot from an LLM is, like, it mode collapses or it's actually not very diverse. And I would expect that you need that diversity to, for that, error loop, like in the star if you have a star like process. So where do you think that's a problem in practice and where do you get the diversity from?

Daniel Franzen 00:22:17

If you have several models, you can just take each thought, embed it like in a vector space and then check if it's too close and then discard it into a new 1, right? Like there are always engineering solutions, but if what you say is true and they just use a single model, then it's just trained very well. Right? Then just good at outputting creative and new ideas in the thinking process.

Michael Hersche 00:22:41

Or maybe having kind of long enough thoughts. Right? Whereby default processes might be noisy and then you might get out of your local units. Right?

Audience Member 2 00:22:49

The way that the chain of thought works right now is just internal, right, internal to the model. But obviously, like, you can do other stuff in while you're thinking. But OpenAI has made the product and competitive decision for now to just keep all that internal, right? So as soon as we add what they call function calling, tool calling, whatever to that loop, is that going to shift the advantage to more open source modulus? Like, what do these systems look like in your mind having built something like it?

Daniel Franzen 00:23:18

My spiciest take is that I don't think that chain of thought is the right approach. So when humans think, right, we have an idea of the sentence we want to speak before we speak it, and then we try to find the words that communicate this idea. And a language model try to externalize memory or thinking into words at a moment. Like the the chain of thought process is basically just we need some type of memory where where we can write down thoughts. And there are some approaches at the moment. I think Meta has a really nice paper, LCM, I think, where they Makes sense. Transform. Yeah, exactly. Where they try to move from the discrete token space into a more continuous space for thinking. And I think if you can get that to scale well, it's a far superior approach when you have just arguing in the concept space instead of having to default to discrete words, you know?

Benjamin Crouzier 00:24:14

You build that word, with tokens.

Daniel Franzen 00:24:17

Yeah. That's it. Yeah. I

Michael Hersche 00:24:20

But but you're still missing kind of state representation. Right? Kind of an internal state. And that way, right, you just feel like to me, doesn't really, like, significantly change something compared to the standard chain of thought. Right? It's just like the way how you present your thought and kind of in a continuous state, but by itself, it's still kind of a Yeah. A chain of tokens or whatever.

Benjamin Crouzier 00:24:44

So I agree with that. And the neuroscience is very clear that in humans, thoughts precedes language. Like, you can lose your language area and you can still do, like, a logic kind of tasks. But, for example, the byte transformer results are very lackluster. So is there something else that we are missing? Yeah. What what does that tell you? Like, is there is that actually a problem? Is what I'm thinking.

Daniel Franzen 00:25:08

We have a lot of language to train on. We have not a lot of thought data to train on. Like, it's just accessible. Like, if you can train on the whole Internet as tokens, how would you even train that? That's the problem, I think. Like, you can't you can't do a normal gradient descent on intermediate steps, you know, and it makes it very hard.

Benjamin Crouzier 00:25:27

Yeah. I guess maybe your best bet is some RL type approach, maybe similar to the 1 where you get to do your own thinking and reinforce the good type of thinking, but that requires a self play loop that is making you improve. And maybe we don't have that.

Audience Member 2 00:25:49

How much do you think we can compress the reasoning output that we're currently doing like outside the model, basically, is feed it back in inside the model? And that seems to be something that SONET 3.6 or something is doing in a way, right? Because in essentially 1 forward pass, it does a lot of stuff correctly that 1 doesn't, even though it has more thinking time. So there must be some translation function rate can map and spend money maybe on reasoning, but then actually training the model on that reasoning. So I guess the question is to what extent do you think that works?

Tim Scarfe 00:26:20

We discussed this earlier. I mean, with your system on ARC, at the beginning, you need to do lots of thinking, but you just continue to fine tune. And over time, it doesn't need to do thinking anymore because it just knows. And I can imagine that the reason why Claude's model works so well is because they've already done something not too dissimilar to this thinking process in their pre training, and it's already internalized a lot of common thinking.

Audience Member 2 00:26:42

I mean, that's the hot take on Twitter. Right? They they they have had Opus, allegedly, for months, but they're just not releasing it because there's no point. They're just using it to improve their cheaper model

Daniel Franzen 00:26:52

Yeah.

Audience Member 1 00:26:53

In exactly that way. It still is

Audience Member 2 00:26:54

not answer to my question, though, which is to what extent does it work? Like, much can we scale that up?

Tim Scarfe 00:26:59

It should be able to after it's done a bunch of thinking. I mean, this is part of my suspicion on why the o 1 and the o 3 model works so well, right? Because everyone's been throwing lots of arc data. I mean, did debate this because they're not technically supposed to use data from the API to fine tune on, so we don't know for sure.

Benjamin Crouzier 00:27:15

What about frontier math? What about frontier math? Oh, it's also good at frontier math. And I don't think there's been too much frontier math turnaround.

Tim Scarfe 00:27:25

Yeah, it's so hard to know with the dataset contamination. By the way, made an interesting comment about open source. So, there's always this question with language models of how much do you actually need the base language model to be able to do things out of the box versus the practicality of fine tuning it to do some specific thing you want it to do. And it turns out for most, you know, stuff that we're gonna build in the real world, we need to set up these complex bespoke architectures where we fine tune it to do very specific things. So then what's the difference between a frontier model and something like, you know, like Llama? So there seems to be like 2 worlds at the moment that there's, you know, the open AI dream that we have this general intelligence that we could 0 shot absolutely anything and it can do thinking versus the reality of building enterprise apps where we need to do a lot of work and it's quite bespoke.

Audience Member 2 00:28:11

Which dream do you believe in?

Tim Scarfe 00:28:12

Well, I would love it if there were such a thing as a general language model that could just do thinking and figure things out 0 shot, developer aware generalization. But I don't know. It seems to fly in the face of a lot of our practical experience.

Daniel Franzen 00:28:29

There's also a crazy time like benchmark after benchmark gets beaten by LLMs. Yeah. But we still don't obviously, like, you can just look at it. It's not what we mean when we say general intelligence. Like, it can do amazing stuff. In in many ways, it's smarter than almost any human I know. Like, The LLMs are good at the stuff they do, but they don't feel like a general intelligence. Right? And I'm wondering if it's possible that we live in a world in a few years where we can't find any benchmark that is not immediately beaten, but we still don't have the general intelligence we want. If that is if that is even possible.

Tim Scarfe 00:29:08

Yeah. That that'll be the sad thing is when o 3 comes out, and we all have this, like, horrible moment where we think, oh my god, solving arc didn't actually make any difference. Like, we're still stuck. The only thing that that I really have updated though because I'm using ChatGPT Pro and it's genuinely unreal. I'm using it every day. And absolutely everything I use it for is just dramatically better than any We were saying like earlier that, you know, the way we we judge the vibes of intelligence is quite relative. So so it makes Sonnet look dumb. And that's insane.

Audience Member 2 00:29:44

Doing what?

Tim Scarfe 00:29:45

Oh, I mean, I can give you a list of things. Know, whether it's generating generating code. I mean, like, last night, for example, I accidentally deleted the table of contents for Chollet. So and I generate it with Anthropic. I I have to tile it, like, recursively. Otherwise, it's just

Benchmarks, AGI Definitions, and Moving Goalposts

Tim Scarfe 00:30:00

terrible. Just 0 shotted it with ChatGPT Pro. It just produced the best table of contents I've ever seen in my life. It did actually hallucinate for other reasons. We talked about that. But but no, but like when I took out the hallucinated lines, it was just ridiculously good. And not only that, you can just stack up, you know, I've always had this mindset of using language models that you have to partition a problem into smaller sub problems because there's a computational budget. There's only so much stuff that you can do. So, know, we have this intuition about how we partition problems and do engineering and so on. With ChatGPT Pro, because it's got like this variable amount of computation, you can just give it 5, 6, 7, 8, 9, 10 things and within reason it will just do all of it and it will just give you the result when it's done. And that's a completely different behavior than we've ever seen from language models before.

Audience Member 1 00:30:42

If you say Pro, what is the underlying model there?

Tim Scarfe 00:30:45

We don't know for sure, but I assume it's the GPT-4o model. And I'm guessing that the o 3 has the base of the Orion model, the 4.5 or the 5 model.

Audience Member 1 00:30:56

So it just says like, ChatGPT Pro, but you don't have a drop down selection of what model and you just get more?

Audience Member 2 00:31:02

It does say o 1 Pro.

Tim Scarfe 00:31:04

O 1 Pro. Yeah.

Audience Member 2 00:31:05

Yeah. That's yeah.

Tim Scarfe 00:31:07

o1 Pro. But the amazing thing is just how the character changes. I think GPT-4o is a dumb model. It's always getting better though, so it's less dumb recently. But, you know, it's it's just It's a really dumb model. I hate the style of it. I hate, you know, like the delving and and so But I just I just think it's a dumb model. And what really just blows me away is how doing this kind of thinking process changes the character of the model. It it feels like a completely different model in character. Even though it's based on something, you know, that we know. It's the same thing, you know, you can add thinking to like, you know, Llama 1B model and it behaves as if it's a Llama 8B model.

Benjamin Crouzier 00:31:47

So how much difference do you observe between o1 and o1 Pro?

Tim Scarfe 00:31:50

Absolutely huge difference.

Benjamin Crouzier 00:31:52

Surprising because they said it's just the same model with small computes?

Tim Scarfe 00:31:56

Yeah. So o1 doesn't want to think. So usually it thinks for about a second. And I'm guessing they're doing that just to kind of get people to upgrade to Pro. So, thinks for a second or 2. But ChatGPT Pro's, I mean, sometimes you can get it to think for about like 12 minutes. And that's actually really annoying when you're programming. But another interesting fact by the way is that part of the reason it seems better at programming, I wonder whether they've done this on purpose because you can't use it through an API, so you can't use cursor. Because most of the failure modes on cursor are the apply model. So when you're actually forced to take the the code yourself and manually copy it back into VS Code. Like it actually works because obviously, you know, the the apply model from cursor hasn't messed up. Another thing is the apply model in cursor is now better. They've got they've updated it with a version that does like 5,000 tokens a second and it doesn't delete your code quite as much, only like, you know, 2 out of 10 times.

Audience Member 2 00:32:48

What's your observed correlation between thinking long and actually better outputs? Like, is there a point at which it starts to just degrade again?

Tim Scarfe 00:32:55

I can't say I've noticed a correlation, but, you know, other than more thinking better. But whether whether there's some kind of yeah. I don't really know.

Audience Member 1 00:33:02

You said you're gonna be very sad if the ARC benchmark is gonna get beaten up. Nothing changes. What do you think the ASI benchmark is gonna be? The new thing for people to be? Is there something already forming in very certain Twitter spaces? You know, like, what's the next cool thing to work on that we can already

Benjamin Crouzier 00:33:25

1 thing I want to say is I would be sad if frontier map gets beaten and nothing changes.

Tim Scarfe 00:33:30

Right?

Daniel Franzen 00:33:32

There are a lot of different problems. Like, obviously, the next ARC iteration is dropping soon. There are a lot of other interesting problems. Like, at the moment, there's a Kaggle contest for AI Math Olympiad, where you can use up to 32 b models. And the only goal is you get the math questions and you should output the correct answer model of 1,000. And I think at the moment, the top scoring model does 25, 24 out of 50 problems correctly, which is good. But I don't think they would beat the frontier math benchmark at all. Otherwise, yeah, frontier math benchmark is very interesting as a, yeah, as a problem space. Someday, Arc version 3. I

Audience Member 2 00:34:26

guess, are we testing the the right things just because they're easy to test? Like, I mean, we're sort of obsessed with ARC because it's been unbroken for 5 years, which is cool because that's, like, the only thing. But, like, is that even testing the right thing?

Daniel Franzen 00:34:39

True. AGI benchmark is letting your model make money on the Internet.

Audience Member 2 00:34:43

Well, is it, though? Because it's just gonna be arbitraged away by other models.

Audience Member 3 00:34:47

Question, what could change? Like, of all the things that we learned doing ARCA, the maybe rotating to verify, doing the 3, which which could actually have an application and other things and maybe open new doors?

Daniel Franzen 00:35:05

I think a direct application to many problems is difficult, but conceptual ideas should work for many problems. It depends strictly on the problem. Like, if you do math questions, you could rephrase the questions. Like, can also flip, like, multiplications or calculations, and then you have an augmentation in mathematics space and check if the answer is still the same. Know? Like somebody else asked, we are also thinking about if DFS approaches are possible in mathematics or in calculation mode. Encoding? Encoding. Yeah. Might be interesting to tell

Michael Hersche 00:35:46

But us

Daniel Franzen 00:35:49

it's not a it's not a 1 to 1 translation. You have to work to get it to work.

Audience Member 2 00:35:54

You said you updated a lot after Eufre. So do you have an updated personal definition of AGI, if you if you even have 1?

Tim Scarfe 00:36:01

Yeah. I mean, I I still agree with Chollet that it's like adaptation to novelty. You know, it's skill acquisition efficiency. So obviously, normalize for your existing prize and and experience. And by that setting, I mean even Chollet, he would say that it has broad generality. I think that he suspects that there's dataset contamination for whatever reason he doesn't want to say. I think that he suspects that when he drops the new ARC version 2 that suddenly the scores will come down a lot.

Audience Member 2 00:36:30

But they already said it's gonna be dropping down to like 30%. Right?

Tim Scarfe 00:36:33

That yeah. That's what he thinks.

Audience Member 2 00:36:35

That's what he thinks. I thought he actually tested it.

Tim Scarfe 00:36:38

Well, he still has access. Right?

Michael Hersche 00:36:40

I don't know.

Tim Scarfe 00:36:40

Yeah. I don't know if he still has access to it. But the thing is, with the dataset contamination, right, the reason why he had such a clear separation between these benchmarks is like, you know, the first time you test a frontier model on on on a certain test set, that's it. It's gone. Data's leaked. Right? So you only really get 1 shot at it. So he probably thinks that, you know, he's got this new set of ArcTasks now. And the first time he tests it on o 3, I'm sure he thinks that it won't work very well. But but then it's lost. And it's so expensive to create these tests. You know, my my cohost, Dr. Duggar, he reckons that there's we should create like a compositional, like a, know, recursively compositional version of ARC, you know, which means that it will have some generative property, So, you know, we can actually create novelty every single time we test.

Audience Member 2 00:37:26

So you and Keith had, when o 1, I think, came out, made a video about reasoning, the 1 where Lex Fridman, you remember?

Jan Disselhoff 00:37:34

Yeah. So

Audience Member 2 00:37:37

I guess 1 of the top complaints about that video was that just the goalposts keep moving. The goalposts said to

Audience Member 1 00:37:44

the Kuiper Belt or something it was, I think.

Audience Member 2 00:37:46

Was a it was a good joke. But, like, what do you think that's a good thing that the goalposts keep moving? Or maybe the better question would be, do you think if we went back like 5 years and presented people what we have today, just a chat GPT interface as it is today, would most people agree that it's AGI?

Tim Scarfe 00:38:02

Well, in many instances, it kind of is. It's a jagged form of intelligence. Right? So there are many things that it does very, very well, and it has some massive holes in it. I think this whole conversation with the o 1 models has made us think more about what reasoning is. You know, I think I had a very binary version of of what reasoning is. And actually, we've come to accept that the things that language models do can be considered a form of reasoning. And there is gradations of reasoning. You know, like the the the more generalization you do, the more parsimonious your model is. You know, the smaller the computational circuits, if you like, the stronger the reasoning. And I don't think it's correct to say that they don't reason because they clearly do. And there's also another thing that Subbarao Kambhampati talks about, which is like there's a lot of symbolic AI people who have this very puritanical, you know, sort of going back to Aristotle and so on, know, like logical syllogisms and so on, you know, like, only Greek reasoning is legitimate. And I I think that that's I mean, there's there's something to be said for that, you know, that there are people who think that we should think of the world, in logic, and that there are certain rules that the world conforms to. And and that's correct, but it's it's not the only way to think about reasoning.

Benjamin Crouzier 00:39:15

I think, we can close on that and, yeah, thank the panel.

References

[1] **ARC Prize**
https://arcprize.org/

[2] **Tufa AI Labs**
https://tufalabs.ai/

[3] **Jan Disselhoff**
https://www.linkedin.com/in/jan-disselhoff-1423a2240/

[4] **Daniel Franzen**
https://github.com/da-fr

[5] **Michael Hersche - IBM Research**
https://research.ibm.com/people/michael-hersche--1