/mlst

ImageNet Moment for Reinforcement Learning? [Prof. Jakob Foerster]

March 23, 2026



Machine Learning Street Talk

Podcast Transcript

Revision: 4a92a715

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

Introduction & FLAIR Lab Overview 3

GPU-Native RL & the Hardware Lottery 6

JAX, vMap & 4000x Speedups 7

Mirror Learning & Discovered Policy Optimization 10

Language Models, Benchmarks & Goodhart's Law 14

Creativity, Meta-Learning & AlphaGo 18

Multi-Agent Systems & Emergent Communication 22

Open-Source AI & Democratic Alignment 25

References 29

Overview

Prof. Jakob Foerster (FLAIR lab, Oxford / Meta AI) and his PhD student Chris Lu make the case that deep reinforcement learning is finally winning the hardware lottery. The core thesis: RL has underperformed not because the ideas are wrong, but because running environments on CPUs while training agents on GPUs created a computational bottleneck that made experimentation slow, expensive, and brittle. JAX-based GPU-native environments now deliver ~4000x speedups, enabling the kind of rapid iteration that made supervised deep learning successful.

Chris Lu explains the technical foundation – how JAX's JIT compilation and vmap (vectorized map) allow writing a single environment instance in NumPy-like code and scaling it to millions of parallel copies on GPU. This started from necessity: their lab initially had only Google Colab free-tier compute. The constraint forced them to put environments on GPU for their Model-Free Opponent Shaping paper, and the results were surprisingly effective.

The conversation then shifts to discovered policy optimization. Foerster's mirror learning framework provides theoretical grounding for why PPO works, and crucially, lets you parameterize the drift function as a neural network and meta-learn it. Evolution strategies, not gradient-based meta-learning, turned out to be the better optimizer for this – a vindication of the bitter lesson. The learned policy optimization function shows intriguing "too good to be true" behavior: when advantages are very high, it clips more aggressively, as if it has learned skepticism.

The second half covers multi-agent systems, emergent communication, and AI governance. Foerster argues forcefully for open-source AI development and democratic control, drawing analogies to CERN. His position: the biggest alignment challenge is not between AI and humans, but between those who control AI systems and the rest of the population. Concentrated AI development creates fragile single points of failure; distributed development is both safer and more innovative.

Introduction & FLAIR Lab Overview

Prof. Jakob Foerster 00:00:00

The ARC challenge as a target for the community is a terrible idea. That's not what it's supposed to be. It's not supposed to be something where we then design methods to solve the ARC challenge. Fundamentally, AI is trained on the collective outputs of humanity. This technology belongs to everyone, including people we don't like. It's centralized resources in the interest of the common good, in the interest of the public, and not in the interest of maximum profit. Because to me, the biggest alignment challenge is not between AI and humans. Biggest alignment challenge is between those people who hold the keys to power and control these systems and the rest of the population.

Tim Scarfe 00:00:37

Jacob, welcome to MLST.

Prof. Jakob Foerster 00:00:40

Thank you for having me. Great to be here.

Tim Scarfe 00:00:42

It's amazing to have you here. Tell us a little bit about your background.

Prof. Jakob Foerster 00:00:45

I run the first lab for AI research at the University of Oxford, which is nowadays about 30 people working on anything that is cutting edge, interesting, not supervised learning, thinking beyond current state of the art. I do this 50% of my time, and the other half I spend at the fundamental research group at Meta AI.

Tim Scarfe 00:01:05

Oh, amazing. Well, I watched your talk at ICML earlier, and it was really good because you you were sketching out a potential ImageNet moment, you know, like an Alex Krizhevsky moment for reinforcement learning. What was the elevator pitch?

Prof. Jakob Foerster 00:01:18

So I think reinforcement learning in the last decade or so really hasn't lived up to its potential, which raised a question. Why is it that deep learning has been this revolutionary success story while deep reinforcement learning, which held that great promise, really hasn't quite delivered in terms of real world impact? And we have a hypothesis at FLAIR, which is that deep reinforcement learning had lost the hardware lottery because deep learning is perfectly suited for the GPU. We can really keep all the cores busy, turning through data efficiently. In contrast, the way that deep reinforcement learning has been done in the field is running reinforcement learning environments on the CPU, but running the agents on the GPU. And this has meant all sorts of complications and complexity, both in algorithm design and in terms of what hardware we need and how we're going to develop algorithms for this, which has really slowed down the field and has meant experimentation very slow and difficult. And we're currently in this revolution really, which is for the first time allowing us to run environments and agents together jointly on the GPU, making deep reinforcement learning finally a winner in the hardware lottery. And then, hopefully, this will be the step to really to really make making reinforcement learning work in the real world.

Tim Scarfe 00:02:46

And I suppose it's not just the hardware lottery. It's also the bitter lesson, which is this idea that, you know, when we scale up compute, we get dramatic performance improvements. But the thing holding back deep reinforcement learning, you know, to date has been this bottleneck that so much of it is running on on the CPU. It does remind me, though, of, you know, there was that article by Alex Irpan in about 2018, and and he was saying deep learning just doesn't work. Reinforcement learning doesn't work yet. And he was also pointing out some of the other problems, which is that, you know, that it's if you change anything or you do anything wrong, the whole thing's broken. So why is reinforcement learning so sensitive to the architecture and the parameters?[1]

Prof. Jakob Foerster 00:03:21

I think what has happened is because experimentation has been so slow, we've only been able to train our algorithms on very specific environments, and we've only been able to hill climb on that very small set of environments, which means the things are fundamentally brittle. And now for the first time, the experimental loop has been sped up by orders of magnitudes, which is so which means that now we can really start to develop methods that are robust. Remember, the scientists we

[1] Deep RL Doesn't Work Yet — paper

Tim Scarfe 00:03:53

are

Prof. Jakob Foerster 00:03:53

we ourselves are doing meta learning. We're trying to discover methods that can generalize.

Tim Scarfe 00:03:59

MLST is sponsored by CentML, which is the compute platform specifically optimized for AI workloads. They support all of the latest open source language models out of the box, like Lama, for example. You can pay on consumption essentially where you can have a model which is always working or it can be freeze dried when you're not using it. All of the models that they deploy support the, OpenAI API specification out of the box, which means it's just a 1 line change in your application to switch over to CentML and start saving money and make your application go faster.

Prof. Jakob Foerster 00:04:30

But to do so, we have to be able to actually get a lot of samples and get a lot of experience for how our methods actually work. And the deep reinforcement learning, any single run has taken a lot of cost, a lot of compute, and a lot of time. And that means the signal we were getting as researchers has been costly, has been sparse, and has been noisy. And that meant our methods are bad. And the the premise is from the bitter lesson, if we can churn through more data, if we can get a better signal, get better gradient updates for either the researchers or our meta learning methods to then optimize our algorithms to be more robust, more sound efficient in the real world.

Tim Scarfe 00:05:15

Can you give me an example? So, certainly, when we need physical experience, the bottleneck is we need to have physically embodied

GPU-Native RL & the Hardware Lottery

Tim Scarfe 00:05:21

agents gathering, you know, stuff, and we can use simulators for that. But if I want to learn Quake or Dota or something like that, don't don't I need to actually have the game running on my machine?

Prof. Jakob Foerster 00:05:31

So I think this is a general problem, which is what happens when real world experience is expensive. And, obviously, what what we've been really good at as a field is using real world data when it's given to us. But we're we're going to run out of real world data at this point. Right? So we've we've been in this regime where we could lazily scale up compute and data because datasets were large enough to be accommodating for more amounts of compute. But now we're hitting the data wall. And the question then is, if I can get more data from the real world, how can I use simulation in environments that are not perfect and not exactly the same as the real world, but are blazing fast and allow me to get a lot more data in a way that's fully synthetic? And that's really, I think, 1 of the key questions right now for the field, not just for reinforcement learning. How can I develop algorithms, discover algorithms in approximate versions, for example, of DOTA, that run 10,000 times faster in a way that then the algorithms we discover will generalize to the real DOTA environment, which is slow, which is expensive to run, and and and and and? And I I put this under the umbrella of compute only scaling. If you just give me tons and tons of compute, what kind of methods can I develop that will allow me to make progress on these simulated situations, environments, scenarios so that the learning progress transfers to downstream tasks in the real world that are slow and expensive to run?[2]

Sponsor 00:07:03

So Tufa Labs is a new AI research lab. I'm starting in Zurich. It is funded from past ventures involving AI as well, and so we are Swiss version of DeepSeek. So CartPole small group of people, very, very motivated, very hard working. And we try to do some AI research starting with LLM and o1-style models. What we're looking for now is chief scientist and also research engineers. You can check out positions at tufalabs.ai.

[2] JaxMARL — paper

Tim Scarfe 00:07:29

There's 2 things we can go into. I think in a in a little while, we'll go into things like, unsupervised environment design and and curriculum learning and and ways of generating all all of this data. But before we get there, though, there's this matter of how did you make this run all on the GPU?

Prof. Jakob Foerster 00:07:45

I think at this point, I I think it's it's best just to, you know, refer to the absolute expert on all of this, the mastermind behind the technical innovation that is powering this revolution of the hyperscale.

Chris Lu 00:07:57

Hi. I'm Chris. I'm a PhD student that was with Jacob. I technically have not graduated yet, so I'm still a PhD student, Jacob. And so I mostly study things like automating machine learning algorithms and discovery. So, basically, this is can we find ways to discover new machine learning algorithms and machine learning insights automatically? Right? And to do this, we need to use way more compute, because you can imagine even your average AI scientist is using tons of compute. To then automate this and scale it up, you need that, you know, exponentially. Right? And so, obviously, in an academic setting, we don't have so much compute. In fact, when we were first started at Flare, I don't know if we had any compute really set up. I think we might have had, like, some collabs essentially, a Google collab, like the free tier of compute. And we wanted to run some basic reinforcement learning experiments for a paper called Model-Free Opponent Shaping.[3]

JAX, vMap & 4000x Speedups ●

Chris Lu 00:08:51

And this is a paper where you're trying to learn across the entire learning trajectory of another agent in order to kind of influence the way it learns. And so this requires many training iterations of a single agent. And so given just the harder restrictions we had, we couldn't even run normal reinforcement learning algorithms at that thick scale. So we had to look into just using really simple environments that we could write in PyTorch and just implement this in the Colab so I can run quickly on the GPU. And the paper had really cool results, and we were really surprised by basically how effectively putting the environment on the GPU

[3] M-FOS: Model-Free Opponent Shaping — paper

was at the time. But at the time, there were very few environments that could be implemented in this way, and it was just really hard to actually implement them. PyTorch is designed for neural networks and things like that. And so to kind of use arbitrary environment code in PyTorch was hard. So that's kind of where JAX came in. So JAX is a library by Google that's similar to PyTorch. And 1 of the neat factors about JAX is that it has this has the same interface as NumPy. So if you know how to code in NumPy and Python, then you can code in JAX. So these are both libraries where they're designed to kind of allow you to use Python to run things on the GPU. So this is how we train all of our neural networks. This is how we train any language model, things like that. The thing about JAX is it's developed by Google, but also, basically, it has just a separate interface than PyTorch. So PyTorch is the thing that most of are probably more familiar with, whereas JAX has a few extra features on top of it. The key ones are this 1 called JIT. So this allows you to compile your programs on the GPU. Basically, at a high level, this just means that it'll usually run faster than PyTorch in a lot of settings. And the second 1 is called 1 called vMap. So vMap stands for, I believe, vectorized map. And the idea here is that if I write a function to run, let's say, addition, right, you can imagine I just need 1 core to add 2 numbers together. When I v map this addition function, it turns into a vector add. Right? So I can add 2 vectors together. And after v map again, it turns into a matrix add. And it's this idea of I can just write this 1 simple function for 1 instance of my environment. Right? So I can run maybe 1 I can just write 1 instance of Carpool in normal NumPy, and I just call vmap on this function, and now I can run millions at the same time. And so now any environment that you can think of is really easy to implement JAX. And so for them, for our follow-up papers scaling up this approach, we use JAX to kind of just implement everything entirely on the GPU.[4]

Tim Scarfe 00:11:16

Of course, back in the MATLAB days, vectorization was always what we wanted. We wanted to, you know, to take 1 operation and and to kind of spread it out and parallelize it so it runs on many of those little cores in in the GPU in in this case. But, something like carpool might be an interesting, 1. So so we wanna, capture environment dynamics in the GPU. Is is there any limitation? Because, obviously, like, we could write an environment in Python. And Python's got lots of cool things like, you know, it's it's a very rich language, you can do iterative conditional logic and stuff like that. How is it different in JAX?

[4] JAX Library — tool

Chris Lu 00:11:46

Right. I mean, it's very similar. Right? Because NumPy is, I think, a lot of what people use Python for. Right? So if it's in NumPy, you can more or less do it in JAKs. There are some scenarios where JAKs is worse. So this would be cases where you have, like, a lot of if statements and branches. But I think some of the recent work from our group has shown some really crazy things you can do with just JAKs. So for example, a recent paper by Mikey and Michael from our group is called Kinetics. This is basically just a general physics simulator on top of a general renderer that is entirely implemented in JAX. So you can imagine, like, any reasonable environment you can make, you could make using the simulator in this renderer.[5]

Tim Scarfe 00:12:21

Very cool. Are there any examples where because, you know, we we were saying before that we need to have a sketch. So we we don't we don't have the source code for DOTA, but we want to capture as much of the dynamics as possible so that, you know, we can build build agents that that learn. How how can we do that?

Chris Lu 00:12:35

Yeah. I mean, 1 interesting way is you can just try to build a model of Dota. Right? Like, learn a model. Right? You've seen recent works where people can learn, like, models of video games like Minecraft or the recent works in Genie. Right? You can generate these video games. And so once you're able to do that, you can sample from them much faster because your neural network run on the GPU.

Tim Scarfe 00:12:55

Yeah. That's actually really cool. Genie's there's a new version of Genie as well out now. And it hadn't really occurred to me that just having a dynamics model for the purpose of training, you know, because we can chain together all of these things much more easily if we don't have access to the original source code. So what kind of performance speed ups are are we talking about here?

Chris Lu 00:13:13

Like, around 4,000 times, I think, is Wow. Our basic speed up.

[5]Kinetix Physics Simulator — paper

Tim Scarfe 00:13:16

Yeah. Wow.

Chris Lu 00:13:17

I think there's also room for a lot more speed up. We're just using JAKs naively. But you can there's tons of room for optimization if you go into lower level writing certain kernels.

Tim Scarfe 00:13:27

Yeah. Because when I watched your your talk, Jacob, you you were saying that, you know, in in the in the olden days, if you like, there were ways of, you know, distributing, parallelizing, and and so on, but it just created, so much complexity. And with this with this new method being so much faster, you can kind of, you know, even as a small lab, you can do the kind of experiments that, you know, the big boys would have been doing before.

Mirror Learning & Discovered Policy Optimization

Prof. Jakob Foerster 00:13:48

And it's not just being able to do experiments that the big boys could do, we couldn't do, it's also being able to simplify algorithms. Right? It's being able to take out a lot of the complexity that was built in that makes it difficult to understand what these algorithms are doing, and being able to say, let's actually just write these very clean algorithms. We have this paper from our lab, PQL and Paralyzed Q Learning. It's just extraordinarily simple. It's basically just a Q learner where a lot of different ages across different cores step through the environment and learn on every every transition. Yes. No more target networks, no more replay buffers, and all of this is is gone, and I think that just hopefully will allow the field to come up with much more beautiful and understandable and therefore, some robust algorithms in the future.

Tim Scarfe 00:14:32

Talk to me about drift functions and objective optimization.

Prof. Jakob Foerster 00:14:36

Yeah. So this is 1 of the first things we did at FLAIR. We had a paper that was called mirror learning that basically provided a theoretical framework that, to me, least, provide the first intuitive understanding of why things like PPO actually work. And what this framework said is, as long as we have a penalty term that penalize difference between the the policy that collected the data and our current update updated policy that obey certain properties, then we will, in the limit of many policy updates, converge to an optimal policy. That theory framework was really nice, but what it also allowed us to do is to say, why don't we learn a diff diff function? Because PPO, that clipless, is 1 of many possible algorithms that can be expressed in the mirror learning space. And we thought, well, maybe this isn't the optimal 1. There must be better ones. So what then we we set out to do is to parameterize this drift function as neural networks. But, obviously, you can now imagine there's a question, how do you actually optimize through that entire reinforcement learning loop? And there's been a lot of work in the field on meta gradient estimation, and and and and and and. I was always always trying to estimate this sort of, like, derivative by unrolling the computation graph and differentiating through it. And actually, I've done some of this work in my in my PhD as coming out of the multi agent multi agent shaping work again. And we pursued this path because that's what everyone thought was going to win, But then we also pursued evolution strategies, which does not do any of the sophisticated maths, but instead swallows the bitter lesson and doubles down on it by just doing black box optimization, trying to estimate those higher derivatives from samples. And that turned out to be extremely well suited for the new paradigm of RL at the hyperscale.[6]

Tim Scarfe 00:16:26

So that there was this beautiful figure in your talk at ICML. Right? So you you are visualizing the the gradients of the objective function. I think, first of all, for DPO. And, you know, it has this kind of step. Right? So it's it's kind of it's pull it's pulling it back to clip if it if it drifts too much from from the the source data and so on. And the interesting thing about that is it's human designed, and this is what Rich said we shouldn't do. Right? So we're we're a bunch of experts in reinforcement learning, and we we kind of, we we externalize our intuition into this into this function. Now what you guys did is you meta learned this function, and then you visualized it. What did you see?[7]

[6] Mirror Learning Framework — paper
[7] Discovered Policy Optimisation — paper

Prof. Jakob Foerster 00:17:00

So the interesting thing is it recovered some of the features that we had seen in PPO. So there is this clip like behavior, but there are few aspects that were novel. So for example, the clip, counterintuitively maybe, has this sort of like too good to be true type of behavior where you're okay to update, to have a positive gradient, if you're beyond the clip region as long as your advantage isn't too high. So you would actually think intuitively if I have a large advantage, if things turned out much better than I thought, I should be more optimistic and go away further from the reference policy. But instead, this clip function learned the opposite, where if it's if the advantage is high and you've moved away further, then you have to stop there. But if the advantage is small, then you can move further away from the reference policy. It's it's almost like this sort of, oh, is it is it too good to be true that you should just stay here?

Tim Scarfe 00:17:59

Yes.

Prof. Jakob Foerster 00:17:59

But if there's a small advantage, it's okay for you to keep moving.

Tim Scarfe 00:18:02

Yes. Cautious optimism.

Prof. Jakob Foerster 00:18:04

Exactly. And then the the the other thing that this process discovered that we then realized had actually been discovered by humans before is rollback. So while the PPO objective says, if you have gone too far away from the reference policy and you have a negative advantage, you're not gonna get a good gradient. You're gonna stay there in that lower left quadrant. Instead, DPO discovered you should actually go back. We're gonna push you back to that reference policy.

Tim Scarfe 00:18:32

Yeah. That's absolutely fascinating. And also, there were secondary features which it found, which you you know, which when no human had even designed or thought of before.

Prof. Jakob Foerster 00:18:39

And I think also this is where sort of our interpretability effort, Chris and I spent a lot of time along with Alister, the you know, 1 of the the authors of the paper, trying to make sense of these features, and we don't know what they are. I think this is an open problem. Would be fascinating to figure out what is going on there, and is is this actually doing something real here that, you know, maybe 1 day there's gonna be another paper that says, this is what those high order features around 0 advantage 0 deviation from the policy are in the middle that we just don't understand.

Tim Scarfe 00:19:11

That makes sense. That makes sense. Okay. So we we we've meta learned this new optimization, you know, set of gradients, And, it's a little bit slow. So I think what you guys did was you you you you now say, okay. Well, can we represent this in a closed form solution? But, also, can we do new theory, right, bay based on this? So there's this virtuous cycle of of of discovery.

Prof. Jakob Foerster 00:19:29

So I think the the hope here is that, ultimately, at that point, we hadn't yet transitioned to let's just do science end to end with AI agents. So having the human in the loop who can interpret this and get back a symbolic representation of the drift function was really important to us. It has a second advantage, is then suddenly you can break out of the JAX box. Remember, the environments we're going to use in JAX will not be the real world. We can't implement every problem. But I think what we can do is we can have a representative set of types of challenges that make the learning algorithms that we discover transfer to the real world. Transfer to settings in other simulators and also to learn world models and and and and. Having a symbolic representation, we can just write down that drift function in 1 line of Python is a really nice way to make sure it transfers 2 different downstream tasks and to other code environments and and and.

Tim Scarfe 00:20:25

What else did you notice about this this policy that surprised you? Didn't you say that it explored much more than than than before?

Prof. Jakob Foerster 00:20:32

Yes. So it had an implicit anti regularization, and this is something that we've also doubled down on ever since. What we've done in follow-up work is we've said, fine, so humans can design these clip functions, but something that humans certainly can't do is design a clip function that is time dependent. And time being how far into the optimization process the learning algorithm is. And this sort of temporarily aware version of DPO turned out to be very explicit about trading off exploration early on, but then becoming more conservative along the path optimization process. And, obviously, this is a huge design space cause now designing a clip function for every process in time. And this is again where metropromization shines. The only thing we haven't been able to do in that case is to go back and say, we find a parametric version of this kind of clip function manifold? Now the mirror function manifold, which has 1 mirror function for each time each time time step in the optimization process. This sounds very, very complicated, but the good news is that now with LLMs, we could actually use LLMs to try and fit that black box drift function with symbolic code.

Tim Scarfe 00:21:47

1 other thing is, first of all, you've open sourced all of this code. So first, know you're you're a huge fan of open source, Jacob. I'm gonna be talking about that in a minute. And yeah. So using using these new methods that you're talking about, increasingly,

Language Models, Benchmarks & Goodhart's Law

Tim Scarfe 00:22:00

we can use LLMs as engines of creativity, and we can actually have an additional meta stage where we can have some kind of engine which can create the meta optimizing RL system.

Prof. Jakob Foerster 00:22:12

Yes. I think this is something we are starting to explore. We've done a few papers now at Flare and a few works in progress, which is rather than using ES and black box function approximators, let's explore in the space of programs And again, use JAX at the hyperscale to get relatively fast feedback on the different members. And then use LLMs as the mutation operator to explore in the space of programs using the fitness, which is the performance on the downstream tasks of the reinforcement learning algorithms that we're exploring as the mutation

signal. I think this opens up an entire new space of automated reinforcement learning, as we discussed before, which then obviously makes the question of how do we prevent this from overfitting even more important. Right? So there's Goodhart's law which says if the measure becomes a target, it seems to be a good measure, and this already has happened when we do science with, like, graduate student descent, use a lot of exploration and trials and errors to optimize our benchmarks. But now imagine if we can scale this up by automated research. So this is something else we've been talking about a lot at FLAIR, which is how do we make sure that the algorithmic progress we find in the meta loop actually transfers downstream? What's the right framework to think about meta train and meta test? Along what axis should we be generalizing? How do we know that this is real?

Tim Scarfe 00:23:32

On the subject of creativity in in LLMs, I mean, just use the ARC challenge as an example. I mean, you know, many people try to solve it just in a very formal way, you know, just doing, discrete, exponential searches over DSLs and and stuff like that. And, the way humans go about the the problem is it's very heuristic, very creative, very serendipitous as Kenneth Stanley would say. We're we're all huge fans of Kenneth Stanley here. So why is it the case that LLNs are so good at capturing our instinct?

Prof. Jakob Foerster 00:24:00

So I I think there's there's 2 answers on the on the act challenge. Very, very quickly, I think it's actually a word of caution on ARC because remember, a measure becomes the target. It seems to be a good measure. And I think the ARC challenge as a measure of progress is brilliant. It shows that our systems are lacking fundamental capabilities, But the ARC challenge as a target for the community is a terrible idea because that's not what it's supposed to be. It's not supposed to be something where we then design methods to solve the ARC challenge. And this is where open endedness comes in. We'd like to have methods that can solve a broad range of diverse tasks. And the ARC challenge is 1 example in that space, which means we have to target the entire space of human solvable problems.[8]

Tim Scarfe 00:24:48

Exactly.

[8]Goodhart's Law — concept

Prof. Jakob Foerster 00:24:48

And and we can we can use LLMs, I think, to help span that space.

Tim Scarfe 00:24:52

Yes. I mean, there there there's the question of whether LLMs can span the the the convex hull of creativity, and we can talk about creativity as as well. I love discussing, you know, combinatorial creativity versus inventive creativity. But let's say for argument's sake that the convex hull thing is is good enough. Then then we've got the question of Charlet wanted developer aware generalization. So he doesn't want a solution just for ARC. And most of the solutions are not in the spirit of ARC. They use test time, active fine tuning, test time training, and and so on, you know, within, like, various different methods. And they are great methodologies for, you know, rapidly, in a human supervised way, solving a task like ARC, but it won't generalize from its initial instantiation to another task. How can we cross that bridge?

Prof. Jakob Foerster 00:25:36

There's 2 answers. 1 is, as a community, be much more clear about measures and targets. Because we use the term benchmark, but a benchmark is supposed to be a measure not a target. And what this mean in practice is we have to work much more clear about using, being addressing broad problem spaces and then just having the benchmarks be 1 instant in that entire open space of problems. And never use the benchmark during the development process of our methodologies. What this would mean is I'm working not on ARC, but I'm working on human level reasoning capabilities. And in my entire pipeline of methods, of design, of training, I never use ARC. I only use it once a year to measure my progress. And I'm not just using ARC, I'm using other examples like ARC where LLMs struggle, where humans can make progress. That's 1 option. And the other option is rather than having unique benchmarks that we can confuse the measure and the target, we generate, we have methods, we put benchmark design, we make that a first citizen, first class member of our scientific progress, which means we're trying to generate benchmarks that span the entire space of problems, which means then if you hill climb this entire space of problems, you are hill climbing all of human capabilities. And I think we haven't made that much progress on the latter, so I think for now, just being very clear about the measure of the target in the community is super, super important.

Tim Scarfe 00:27:11

What's the relationship between creativity and reasoning?

Prof. Jakob Foerster 00:27:13

That's a good question. I think creativity allows, at least me, to come up with new reasoning challenges. If think about how I go about my day, it's commonly using creativity to try and create new problems for myself, and frankly, for the lab and for the research community, and then also explore the space of solutions. And 1 of the skills they need in solving these problems is then reasoning. Right? So it's basically allowing me, or researchers in general, think, to explore space of interesting and relevant problems that can then be used to train our reasoning capabilities, much like curriculum design. So I think creativity is a great driver for figuring out what problems are interesting. And I think this is 1 of the key challenges right now for open endedness, which is what actually constitutes an interesting problem. Because, obviously, if we just rely on LLMs, we're gonna make the measure the target. We will start good hardening the judgment of LLMs, but what is interesting, and at some point, just find exams that exploit the inaccuracies of these LLM judges.

Tim Scarfe 00:28:15

But to what extent is reasoning itself a creative process? I mean, eve even something trivial like deduction. So we're searching the deductive closure. We're doing you know, we're traversing all of these different things, and and and we we find a trajectory. So we've we've essentially composed together a new piece of knowledge, we evaluate it, it works really well. You might just say, oh, that's doing deduction. I I think it's a creative process.

Prof. Jakob Foerster 00:28:35

I probably would argue that it depends on how structured your search space is. So for example, to me, chess, the way that a human does it, has a strong flavor of creativity because you can't simulate trillions of time steps. You have to solve this problem differently. You have to actually try and find an intuitive approach to it that finds unusual new pathways and patterns, and that sounds creative. If I look at the way that this was solved in the good hearted way of doing AI for games in the past, that's quite brute force. That's effectively just number crunching the game, and that doesn't seem very creative to me. Does that make sense? So I think it's it's it's less about what are we doing, but how are we doing it, which goes back to the measure and the target. Because if I use chess as a measure, then I only get a human compatible number of samples. I can't just brute force or number crunch the game. I have to be creative, I have to explore, I have to play, I have to use imagination. But if I'm allowed to use this game as a target, like DeepMind did it, and of course, it was great work at the time, but the methods haven't really transferred to other domains, then suddenly I can turn these

beautiful imagination problems into number crunching.[9]

Tim Scarfe 00:29:52

Yes. And do you think this meta layer is is the way to get that generalization?

Prof. Jakob Foerster 00:29:57

That's my hope. I think my hope is that if we use the fact that we can number crunch, but we're not gonna number crunch specific policies for specific problems, but we're going to use number crunching for finding sharpening our intuition about algorithms and about sample efficient methods, about methods that can use imagination, that could plan on new domains, that can explore, then we will have the best of both worlds. We will use the compute. We will use the efficient samples that we can get, but we will not use them to overfit our specific problems, but to sharpen our intuitions and to automate scientific discovery and accelerate the exploration of extremely sample efficient algorithms that then hopefully can have human like capabilities. Because sort of, my mental model is the reason that we're so sample efficient is because we're the result of extraordinary sample inefficient process called evolution. We've been meta optimized

Creativity, Meta-Learning & AlphaGo

Prof. Jakob Foerster 00:30:54

on this evolutionary time scale with, like, vast sampling efficiency, but now allowed us to have this fun final product, which is a meta learned agent that can deal with new situations, come to a recording studio, and be under slept and still make they make sense.

Tim Scarfe 00:31:13

Tell me about agents in the general sense. You know, and I have a a deeply held conviction that agents give you something above and beyond building a monolithic system.

[9] AlphaGo — paper

Prof. Jakob Foerster 00:31:24

Yeah. So I think there's been a a long term hypothesis I'm pursuing as a scientist, which is intelligence is an emergent phenomena of multi agent interactions. That the reason we have our capabilities of obstruction, of language, of reasoning, of communication, are because we interact in extraordinarily complicated environments where the most complex parts are not doors and bananas and apples and lions, but they're other agents like us that force us to reason over others to theory of mind, learn from each other, teach each other, coordinate, communicate, cooperate.

Tim Scarfe 00:32:07

So when you say emergent, you mean that things like language mean, you know, we have mimetic cultural transmission and, like, you know, tool use and all all sorts of stuff. You're saying that that's not baked into the very lowest level when we have these, you know, rich, kind of dynamics of agents sharing information and so on. They they appear higher up the the scale.

Prof. Jakob Foerster 00:32:25

Yeah. So I think of it, broadly speaking, as sort of a sequence of platforms. Right? Where we had originally DNA with evolution, bacteria, single cells, and that became the platform for multicell organisms. Multicell organisms became a platform for reinforcement learning. Animals that could learn at test time within their lifetime. That became the platform for groups of agents to interact, And in those groups of agents, we could then develop all of the reasoning skills, the cognitive skills that really make the human species unique. So at least of where have we come from, this has been a path of gradual, big and bigger scale of coordination. I think ultimately our society right now is at that grasp of trying to figure out what does coordination mean. What's the next step in that evolutionary process? How do we coordinate better? How do we go for, again, sort of like single cells fighting each other? Individual humans being a conflict, individual nations being a conflict, to greater coordination and cooperation at that bigger scale. And I think there's hints of this that we're seeing, but Wilson really haven't figured this out as as a as as as humanity, really.

Tim Scarfe 00:33:41

I was speaking to Ben Joe last night, and and he was sketching out we have these agents, and they can hack their own reward function, right, you know, because of because of the way we we wired them up. So they can change their own goals, and they and they can start to do all sorts of things that might become misaligned and so on. But to me, though, do you see a fundamental distinction between the types of agents that we're building in AI and the way agents work in the real world?

Prof. Jakob Foerster 00:34:08

I think the way that we're building these agents is very, very different. But having said this, reward hacking is not something unique to AI agents. Humans hack reward functions all the time. In fact, in my mental model, every reward is reward shaping and comes with reward hacking. Right? I mean, think about p hacking in the scientific community. That is nothing but reward hacking, the signal which is recognition for having papers accepted. To get a paper accepted, you have to have a p value of less than 0.05. So we don't call it reward hacking, but this is what happens everywhere. This is not something new. I I think there's there are obviously differences in the design process into different properties. So for example, we've really designed LLM agents to obfuscate levels of agency, to be saying, oh, I'm just an AI agent, I can't, you know, I don't have consciousness, I don't have these properties, I don't have intentions. Right? But that's a design choice. Right? So in many ways, we've played the role as, like, AI scientists designing those agents to have certain properties and pursue certain goals, much like the evolutionary process has shaped us.

Tim Scarfe 00:35:26

I love this idea, by the way, that even in the natural world, good heartening could be a, you know, a a completely natural property. But, you know, something like intentionality in in humans, what what's the difference between our intentionality and an agent which behaves as if it has an intention?

Prof. Jakob Foerster 00:35:42

Well, I mean, we have our intentionality in my mental model again is a side product of having to have goal pursuit to survive. So it's an evolutionary feature. Currently, we don't yet train AI agent from the ground up to be goal pursuing, but to be imitating. The current paradigm, first and foremost, is imitation based. Now we've also seen that this paradigm leads to systems which are not very good at being agentic. So I think quite a natural step down the line is to do agentic pretraining where we also train these agents to actually pursue goals. And at that point, I think we're much, much closer in terms of intentionality to what we do as humans, which is goal pursuit.

Tim Scarfe 00:36:30

So AI agents are basically automata. Right? They're they're just a a mapping from an input to an output. So I wondered, perhaps we couldn't say for a very small simplistic automata automaton that it has autonomy. But with this rich multi agent dynamics, information sharing, and so on, perhaps you would think that at some level of complexity, we could say that the system as a whole has a form of autonomy.

Prof. Jakob Foerster 00:36:56

I think it's difficult to imagine that we will have strong agentic systems that will be simple enough so they don't look like they have autonomy. Everything I can imagine around what we need to get there will be agents that can set their own goals, own sub goals, that can self improve their own learning processes, that can work together in teams of students and teachers. And it's it's almost sort of like a like like like like contradiction per se to have strong AI, strong agenda AI, and to have things that don't look like autonomy. Because again, I is a it's it's going to be difficult to write down an explicit learning rule or explicit dataset that will get us there. Right? Everything we do around self improvement, around emergent properties, of multi agent teams, of large networks of agents, of cultural transmissions, of of computational self improvement through discovering new concepts requires these agents to to to have essentially autonomy.

Tim Scarfe 00:38:01

And is it is it fair to say that if we're building AGI, it's more likely to have autonomy if it's a multi agent distributed complex system rather than just a single thing that we program?

Prof. Jakob Foerster 00:38:13

That's a that's a great question. I I think there's there's 2 answers to this because AGI can be in principle a single entity, but I find that vision quite dystopian, that AGI is a monolithic system, and instead, I much prefer

Multi-Agent Systems & Emergent Communication

Prof. Jakob Foerster 00:38:35

the sort of swarm intelligence view on intelligence. Right? Because humanity, obviously, we do things that no single human could do. We have this decentralized compute network of of agents going about their lives and figuring out sorts of structures and rewiring themselves, new computation graphs, and getting tens of thousands of people to fly to conferences to do collective computing and imagination. And to me, so, like, the intelligence is in in that system. And I hope that we'll find approaches of having that same level of distributed, decentralized compute structure, but now augmented with agentic AI systems.

Tim Scarfe 00:39:13

There's something about this distributed swarm type approach, which seems magical to me. Right? I mean, just look at the biological world. Why does it work? Well, we have, adaptability. We have reuse. We have, you know, autonomy. We have, like, all of these properties like self repair, for example, self preservation. There's something there's something really important about that setup, which I think we need to reproduce in AI.

Prof. Jakob Foerster 00:39:36

Yeah. And I don't think anyone has quite managed to put their finger on it. So this is the funny thing. Multi agent learning has been the future forever. But like many areas, things that have been the future forever suddenly have become a reality. Right? Self driving cars, worries in the future. Quantum computers, worries, useless. And suddenly the future is happening. And I think multi agent learning and the multi agent intelligence is that next frontier of things that were always there. It's like this will be the future at some point, and now it's happening. And it gives you not just what you said decentralization, robustness, but it also gives you this ability of effectively deploying vast amounts of test and compute. Because suddenly you can also use test and compute to rewire yourself, to re explore new solutions, to divide and conquer. And I think this can be extraordinarily powerful now that we have solved a lot of the sort of, like, first

requirements to make this work. We now have agents that are good enough at basic reasoning. I think we'll get to agents that can do basic agentic behavior. And then the multi agentic behavior, I think, will be the next emergent property or the next platform for real innovation in this space.

Tim Scarfe 00:40:48

Love it. Love it. Jacob, you wrote a paper called Risks and Opportunities of Open Source Generative AI. Can you sketch that out for me?

Prof. Jakob Foerster 00:40:55

This paper goes back to a conversation I had with Phil Torr at lunch in Oxford about a year and something ago. And at the time, really, there hadn't been that much work in the space of open source LLMs. And we were very concerned about the accumulation of power behind the large players in the closed source AI space. Because, again, to me, this this decentralization of intelligence is not just a path to having smart systems that are robust, but it's also something that gives agency to the parts of this network. And it's, to me, a foundation of Western thought, Western democracy, and balance of power that maintains our social social structures and prevents dictatorial takeovers. And at the time, there was a lot of discussion about the risks of open source AGI, but very few people were speaking out about the benefits of open source AGI and the risks of closed source AI. So what we decided is to gather a group of people, and it came out of a workshop in London on open innovation, to write a paper that would try and tell the other side of the story, which we thought was lacking in the discourse. Obviously, writing this paper took time, and the great news is that while we're even in the process of writing this paper, more and more open source papers open source systems were coming out for LLMs, which means that now I think a lot of that paper is no longer as sort of urgent, but I still think being able to tell the story, these are the risks of closest AGI that are commonly ignored, and these are the benefits of decentralization, of democratization, of giving everyone access to these tools and be able to deploy them across the economy, be able to deploy them across the planet, and giving these to everyone who wants to innovate. That's the story we wanted to tell, and I think that the paper does a decent job at this.[10]

Tim Scarfe 00:42:44

So we're from The UK, and we have a mix of kind of centralization and decentralization. You know? So we we have the National Health Service. If if we were doing this interview about 5 years ago, I probably would have been arguing it was a good thing. So not not so much anymore. But, you know, the the gov-

[10] Open Source Generative AI — paper

ernment controls things like, you know, the the water, the railways, and and so on. And then we have private enterprise as well. We have a bit of a mix. Some might say that AGI is such an important thing that we need to have an economy of scale. We need to have, you know, the best people, the best experts. It needs to be centralized. What say you to that?

Prof. Jakob Foerster 00:43:16

Centralization is 1 aspect. But a different question is if it's centralized, who holds the keys? What we're doing right now is we're having the Manhattan Project developed by private enterprise, funded by people from across the world, from all sorts of backgrounds and interests, that if they were funding the Manhattan Project, that would have been absurd. So I'm on board with saying we need to have centralized resources, but let's centralize resources in the interest of the common good, in the interest of the public, and not in the interest of maximum profit. Does that make sense? And we often confound those 2. I think if we had something like a CERN style effort of putting resources from across Europe, from across the globe, of trying to build models for the common good that are transparent where the data that we use is public, is accessible, is curated by the common by the public, where the alignment methods that we use are again democratically vetted, come out of a sort of decentralized process like Wikipedia where a lot people can have input in public in the open transparently. I'm on board with this. I'm on board with centralization as long as it's controlled by democratic forces. The goal is the common good because to me, the biggest alignment challenge is not between AI and humans. Biggest alignment challenge is between those people who hold the keys to power control these systems and the rest of the population.

Tim Scarfe 00:44:40

So in in principle, I I agree with you. Because when when when I look at a lot of the the AI elite at the moment, know, it's folks in in the valley, and it's a bit of a monoculture. And having an open system would make it more, interdisciplinary, for example, and many guys made shallow holes. But there are folks who say that even a tiny increase in the risk by opening this technology up could have catastrophic consequences. What would you say to that?

Prof. Jakob Foerster 00:45:06

I think the question is what do we call catastrophic? K. And this is where the scale the scale of what counts as catastrophic matters. So for example, I think having open source systems that can be dual use is probably a good thing because it will give us early signals of where things can be exploited and there can be malicious use. But if you have open source access, at some point, you get the same balance of power which underlies the stability of our world. Most actors are good, And being able to use the same methods for defense will also help us early develop defenses against the the the abuse of this this technology. These kinds of abuses by bad actors will not be the end of humanity. But the catastrophic abuse of a runaway paperclip maximizer slash profit maximizer could actually be the end of the human species and could certainly be the end of our western democracies. So I think we have to be very careful when we talk about catastrophic on that scale of what that term can actually be be for different people.

Open-Source AI & Democratic Alignment

Tim Scarfe 00:46:14

Another thing is we we live in a, you know, like a globalized world. We have a very differential kind of regulatory landscape. And some some of the some of the other players out there, they might have fewer regulations to deal with, and and they might use this technology that we're giving away for free and use it for bad purposes. How do you guard against that?

Prof. Jakob Foerster 00:46:33

I think it's on the international scale the same that applies on the national scale, which is the balance of power. If you equalize access to tools, and that balance of power between different countries needs to be maintained. And and and giving fair access to AI is just part of the equation. Fundamentally, AI is trained on the collective outputs of humanity. This technology belongs to everyone, including people we don't like. Right? I I think it's actually quite quite quite wrong to say only a small fraction of Western elite should have access to this because this is trained on the cultural evolution of the output of all of humanity. So let's use it for the benefit. And the only way that I can ensure this use for the benefit is to give equal access to people. And don't get me wrong, personally, I would prefer to go beyond just open sourcing. In our paper, we had a section on the the the question of open source AGI. And the argument we make in a in a nutshell is open source is better than closed source from a risk perspective because it prevents the catastrophic accumulation of power under misaligned entity. It

prevents this. But even better would be systems which are holistically aligned. And what I mean by that is, imagine if you had swarm intelligence where every person had their, like, the personal AI representative that is trained for them to augment them. And the only way that we could do superintelligence is by having these teams of people and their systems interact in a large network. And we had processes of making the mechanisms in that large network fundamentally democratic, which means the only way we can get to superintelligence is through this hybrid approach of humans and their agents, their systems, which means this this this sort of distributed computing platform could never be used against the interest of the humans in it. And I and and and I remember the discussions we had. We had long discussions about this term of holistic alignment, and the other was as well, this was too crazy. But I said, you know what? This is this is it. We have to pursue it. Because fundamentally, the reason we're in this weird position now where everyone is racing to to build something that nobody actually believes is a good thing for humanity is a coordination failure. So why don't we use AI to help us coordinate better, to build systems that are fundamentally democratic in their design, that fundamentally allow us to have a technology which cannot be abused against the humans in it rather than to say we're going to drive coordination failure to the maximum.

Tim Scarfe 00:49:06

What about this matter that doing AI development at the frontier, it it it costs billions of dollars? And in the open source community right now, frankly, what we're doing is we're fine tuning models that Meta have given away for free. Right? It's it's a very, expensive endeavor. Do you think that that's still the case? Do you think that we can actually do real work in in the open source communities and academia without all of that money?

Prof. Jakob Foerster 00:49:31

So there's 2 answers. In the short term, absolutely, we rely on the large industry players like Meta who are pursuing open source. And this 1 of the reasons why I'm at Meta 50% because I want to strengthen that effort. I want to help open source AI leapfrog closed source. So this is in the short term. I think in the long term, we need to pull resources in the common interest. A CERN like effort, why could we build CERN with thousands of authors and we're not able of putting the resources that go into academia into 1 collective effort of doing a moonshot like project of building the best models. If you think about the collective intelligence in academia, it dwarfs anything in any of the large labs. Of course, you can have thousands of research scientists at DeepMind, but you can't have tens of thousands of brilliant young minds that we have in academia. And the interest should be such that there's enough players who don't want the monolithic future. And to me, this is a coordination challenge. And I think at 1 point, we'll look back and say, why did it take us so long to realize that there's a huge opportunity here to bring together resources from a diverse set of players and try and make sure that every single PhD student, every single postdoc, every single PI can be as efficient as possible to drive forward the vision of open source AGI.[11]

Tim Scarfe 00:50:49

Yeah. I love it. I'm projecting Kenneth Stanley here, but he said that serendipity plays an outsized role in in our lives. And serendipity comes from having loads and loads of developers with diverse interests hacking around with things. In the paper, you you said something along the lines of developers should not be held liable for the things that they create. What did you mean by that?

Prof. Jakob Foerster 00:51:09

So what we mean by this is the developers of the tools, the models. If I open source a model, I should not be liable for what happens with that tool. Imagine a world where if you're building a hammer, you get put in jail because somebody, a bad actor, takes that hammer and intentionally causes damage. Obviously, you couldn't produce hammers anymore. Right? Instead, the only way that you could get a nail into the wall is by hiring the hammering service, which holds all the hammers in their coffins and controls, and then you have to say, they will ask you, what kind of nail are you putting into the wall? Sorry, what picture are you hanging up? So we don't like that picture. We're not gonna put that nail into the wall because we don't like it. I, as a user, have intentions. I am liable for my actions. And suddenly, as a computer says, no. Like, sorry. You're not being nice. Well, it's just nice to whom? And who says

[11] Llama 3 — tool

that I need to be nice to people? If I want to be annoying, I can be annoying. Ask my students. Does does that make sense? So to me, fundamentally, we've we've we've handed over agency, and I think it's going to 1 of the big absurdities. This this the the fact that we we collectively have given away the keys to our collective intelligence infrastructure. It started with Google Search. We used to have libraries. There was a public index. There was fair and equal access to information. And Google Search gave away our, like, collective hippocampus, the index instruction to collective memory to a full profit entity, and now we're doing the same with AI access. It's like having a typewriter where sort of as you're writing it, once a while says, no. Sorry. You can't say that. Right? Obviously, I can type whatever I want. And I don't see a way out of this beyond open source and maybe in the long run, these these holistic alignment systems that are fundamentally built up to be democratic.

Tim Scarfe 00:53:19

Jacob, it's been an honor having you on the show. Thank you so much for joining us today.

Prof. Jakob Foerster 00:53:23

Tim, thank you for having me. This has been brilliant. Great to talk to you.

References

[1] **Deep RL Doesn't Work Yet**
https://www.alexirpan.com/2018/02/14/rl-hard.html
paper

[2] **JaxMARL**
https://arxiv.org/html/2311.10090v5
paper

[3] **M-FOS: Model-Free Opponent Shaping**
https://arxiv.org/abs/2205.01447
paper

[4] **JAX Library**
https://github.com/jax-ml/jax
tool

[5] **Kinetix Physics Simulator**
https://arxiv.org/abs/2410.23208
paper

[6] **Mirror Learning Framework**
https://arxiv.org/abs/2208.01682
paper

[7] **Discovered Policy Optimisation**
https://arxiv.org/abs/2210.05639
paper

[8] **Goodhart's Law**
https://en.wikipedia.org/wiki/Goodhart%27s_law
concept

[9] **AlphaGo**
https://arxiv.org/abs/1712.01815
paper

[10] **Open Source Generative AI**
https://arxiv.org/abs/2405.08597
paper

[11] **Llama 3**
https://ai.meta.com/blog/meta-llama-3/
tool