SPEAKER: Tim Scarfe
So I've just picked up Neil from the train station and we are in beautiful Maidenhead. It's a sunny day so we're going to do some filming today. But yeah, it's a pleasure to meet you sir. And can you please tell us what's going on in the t shirt?
SPEAKER: Neel Nanda
Ah yes. This this is an intricate story of alignment meme law. So there was this beautiful meme where you draw ChatGPT as a shoggoth an eldritch monstrosity from Lovecraftian horror fiction with a smiley face on top. Because language models are bizarre and confusing things that are just I don't know, they're kind of a compressed version of the entire Internet that will do bizarre things and bizarre situations. But then OpenAI tried really hard to get it to be nice and gentle and a harmless assistant and look so normal and reasonable and safe, which is the smiley face mask on top of the underlying monstrosity. But unfortunately the smiley face mask. Many people don't realize how weird language models are. And then this t shirt is a further iteration where you split it open, you zoom in on a tiny, tiny neuron and you try to interpret it and you observe grokking, which is a graph taken from this OpenAI paper and also a paper I wrote on Grok. So this also doubles as the first time anyone's made fan art of my papers and I greatly love this t shirt.
SPEAKER: Tim Scarfe
So we have in our midst Neil Nanda. He says on his website that he's interested in a lot of things but especially maths, truth, excitement, improving himself and improving the world. He says he considers himself part of the effective altruism and rationality communities and we might touch on that later. As you all know, I have a few reservations around X-risk. He's now all over mechanistic interpretability and anthropic, worked closely with the legendary Chris Oehler, the one and only. It's been my dream to get Chris on the show since the very beginning and I hope it actually happens one day. Now Neil, he read Pure Maths at Cambridge University and I can tell you without hesitation he's one of the smartest people we've ever interviewed and I'm saying that before we've even interviewed him. Um, so anyway, as people know, being smart is the single biggest factor in being selected for MSTs. So I've got impeccable taste. That's why you've subscribed everyone. Anyway, before we start, Neil, could you give a shout out to Luna? She lives in the substrate of Discord. She's a huge fan of your work. She pointed me in your direction, and I think she would absolutely lose her shit if you said hello to her.
SPEAKER: Neel Nanda
Hello, Luna. Thanks for introducing us. I'm excited to see how much of a mess this episode turns out to be.
SPEAKER: Tim Scarfe
Amazing. If it wasn't for Luna, this meeting would not have happened. Well, why don't we just while we're still out in the forest and we have the esthetics? What is mechanistic interpretability hitherto known as Mech interp? Because it sounds cool.
SPEAKER: Neel Nanda
Yeah. So. All right. So mech interp is so neural networks are notoriously an inscrutable pile of linear algebra that is a mysterious black box that makes no sense. And the foundational assumption of mech interp is being like, Nah, let's assume that it actually makes sense and there is some human comprehensible structure inside, but that the model has no incentive to make the structure legible to us. It's learned real algorithms to do the task, but the algorithms, the algorithms are stored internally in some weird formats and mech Interp is the study of how to reverse engineer these algorithms kind of analogous to how you might take a computer program that's been compiled into a program binary and try to reverse engineer

that binary into source code. And yeah, the field is young but rapidly growing and full of all kinds of deep confusions and fundamental open questions and low hanging fruit that I would love anyone listening to this to come and try to answer, especially if you hear me say something and you're like, that seems bizarre or like it's missing this obvious thing. Maybe we are. I would love anyone to come prove me wrong.
SPEAKER: Tim Scarfe
And what is the biggest open problem in mechanistic interpretability?
SPEAKER: Neel Nanda
Yeah, so we'll get to this more. But I'd say the current biggest problem is the phenomena of superposition. One of the things that was a really nice assumption of early mechanistic interpretability is that models stored as many features as they had neurons. Each feature corresponded to a neuron. They were all orthogonal and it was beautiful and clear and you had the right units of analysis and models are such. Dimensional objects that it's really important to be able to decompose them into individually meaningful things that can be analyzed in isolation. And it's so useful to know what they are. Superposition is this phenomenon where models realize they can compress more features than they have neurons or more features than they have dimensions by representing them as almost orthogonal things and being willing to tolerate the interference. And this is really important and really annoying because it means that we can't just like look at individual neurons.
SPEAKER: Tim Scarfe
So Neil, you've got this great piece that you posted all about getting started in Machine Terp. Could you give us the sales pitch on that?
SPEAKER: Neel Nanda
Yeah. So I think the sales pitch is machine learning is really boring. Machine learning involves training, an inscrutable pile of matrices, figuring out guessing the right hyperparameters, throwing in a soup of data and seeing what happens. Mechanistic interpretability is all about taking a trained model, screwing around inside and trying to figure out what's going on. And one fun benefit of this is that it's just not that hard to get started. You don't need that much compute. You don't need the trillions of GPUs people are using to train GPT seven. You can just take a model like GPT two poke around in a CoLab notebook, get fast feedback and get your hands dirty and getting to the point where you can actually contribute original research to the field is not that hard. And the point of this post is just saying, Hey, this is not that hard. B Here is just a bunch of concrete things you can do to get to that point along with a bunch of resources and things. You should go check out ways you can learn and a bunch of advice for common pitfalls. Like people often think they need to spend two months reading every possible paper that could be relevant to McIntyre before they can start writing any code. And you can just start writing code basically immediately. Um, and there's just a lot of low hanging fruit in the fields that I think people can come and contribute to.
SPEAKER: Tim Scarfe
So Neil, you said that McIntyre is akin to alien neuroscience understanding the brain of an alien organism. What did you mean by that?
SPEAKER: Neel Nanda
Yeah. So that's a very fun and grandiose way to put it. Um, to be very clear, I mean alien in the sense of weird and unfamiliar not alien in the sense of this thing is intelligent and we can go understand it. No, I don't think Gbp4 really reaches that bar, um, let alone weaker models. So what I mean by this is that when I think about the world, I have this abstractions and way of thinking about things and processing concepts that make sense in my head. Like I think

about an algorithm and I think about it in terms of how I'd write Python code or just in terms of how a how an algorithm would work. But this is not inherently the only way you could think about something like this. And what I mean by alien neuroscience is that models have their own abstractions and ontologies for thinking about the world and representing their features both in terms of which bits of the model represent these, what kinds of algorithms they learn, etcetera. And we'll get into a bunch of these what are called motifs later on the episode. One particularly fun one is this work I did called progress measures for Grokking via mechanistic interpretability. I was reverse engineering. How a model did modular addition obviously modular addition. You add things and then you round off at the end you know adding you do an arithmetic processing unit. It's kind of chill. It's a very simple thing. You add digits and carry. Nope, the model was doing this galaxy brain thing where it decided modular addition was best thought of as rotations around the unit circle because composing rotations adds the angles and circle gets you modular for free. And the model then decided in an anthropomorphic way decided that it should represent the inputs as these Trig terms of various frequencies use Trig identities to compose them and then do some further galaxy brain stuff to extract out the right answer. And this is both. I basically think you could not have reverse engineered this thing without having that insight. But also you can get these insights without knowing in advance what you're looking for. Like I found this algorithm by reverse engineering. But you need to take a biologist's approach, not an engineering approach. You need to look inside the model, poke around, be curious, have the potential to be surprised and see what underlying structures there are.

SPEAKER: Tim Scarfe

Okay. So a very kind of common critique of machine interp is that it doesn't scale to real world huge mongoose models. What would you say to that criticism and can you provide an example of where that's not the case?

SPEAKER: Neel Nanda

One example that we're going to get to later is the phenomena of induction hats. In this paper I was vaguely involved in led by Katherine Olsen called In-context Learning and Induction Heads. We showed that this small circuit we found in tiny models occur seemed something like it seemed to occur in all models. We looked at up to 13 billion parameter models and it seems to be deeply tied to the phenomena of In-context learning where models can track long range dependencies in texts which is behind things like few-shot learning. And and no, I think this is a real insight we got from playing around with toy models that seems to have taught us something important about real models and the DeepMind Antwerp team will hopefully at some point have out some work understanding a circuit behind how Chinchilla engages with multiple choice questions And to me, the thing holding back the field is less scaling and it's more that we just don't really know what we're doing. And there's all kinds of fundamental problems we're trying to work out like how does a model represent its thoughts? How can we decode things from superposition? Do all of our many assumptions and frameworks about models actually hold up in practice? Because for the most part no one's checked very hard. And I know my personal bet is that if we got to a point where I believed we could confidently reverse engineer GPT two small 100 million parameter model. My guess is that scaling that up would not be that hard or would be hard, but it would be much more of a very parallelizable engineering challenge. And in some sense what I'm saying is scaling is hard, but all this other stuff you're not thinking about, it's even harder, which should maybe be discouraging. But and no, I think it's a bet worth making. Yeah, maybe just one thing I'll add for a more emotive pitch is we live in a world where we have computer programs that can do things like speak English at a human level, draw a unicorn in

ticks, write poetry and explain jokes. And we don't really know how we would write a program like this that could do that stuff. And this just kind of offends me. I believe it is possible to understand the underlying structure and algorithms behind this, but very few people are trying and looking. And it just seems to me that as we're in a world where machine learning is a field where we stack more layers and we make models better, but we don't really know how and it very much feels like an engineering discipline to me, not a science. And my vision of McInturff is I want that to be a thriving subfield of machine learning that is really understanding the internals of these systems and the beautiful emergent structure, but more in the manner of a biologist looking at what's out there and trying to study and engage with it than an engineer building a better system or a mathematician trying to prove theorems that may or may not actually engage with the underlying ground truth. And there's something I find beautiful about trying to be a truth seeking biologist to a system that is ultimately made out of maths and where I know exactly what's going on on a mathematical level, but I don't know what any of it means. And if that speech resonates with you, I'd love for you to come join me. But yeah. Shall we wander on to the recording?
SPEAKER: Tim Scarfe
I think we should. Yeah. That is a question. Right. Nick, eat the path, he says broad question. Do you see McInturff as chiefly theoretical or an empirical science? And will this change over time?
SPEAKER: Neel Nanda
Yeah. Um, I see this as very much an empirical science with some theory sprinkled in, but you need to be incredibly careful. So fundamentally, I want to understand a model and I want to understand how the model works. And a sad fact about models is models are really fucking cursed and just work in weird ways that aren't quite how you expect and which represent concepts a bit differently from how I expect them to and just do all kinds of weird stuff I wouldn't have expected, like when I'd poked around inside of them. And I think that if you're trying to reverse engineer a network and you don't have the capacity to be surprised by what you find, you will not doing real mechanistic interpretability. It's so easy to trick yourself and to go in with some bold hypothesis of this is what the network should have and you probe for it and it looks like it supports that when you dig further and you are wrong. And yeah, I think there is room for theory. I think in particular we just don't have the right conceptual frameworks to reason about how to understand a model and we'll get into fundamental questions like superposition later on. But yeah, I think that theory needs to come second to empiricism if your theoretical model says X and the real. Says why your theory was wrong, which is the story of all of machine learning.
SPEAKER: Tim Scarfe
So goji tech, she says. Question for Neil. Does he think a foundational understanding of deep learning models is possible and does that extend to prediction using a mathematical theory?
SPEAKER: Neel Nanda
Possible is such a strong word. Like if we produce a superintelligent AI, will it be capable of doing this? Probably in terms of foundational understanding. Um, I think there are deep underlying principles of models. I believe there are scientific explanations for lots of the weird phenomena we see like scaling laws, double descent lottery tickets, the fact that any of this generalizes at all. I'm hesitant to say there's some like strong things here or some strong guarantees like, I don't know, models are weird. Sometimes if you change the random seed, they will just not learn. I'm pretty skeptical of basically all mathematical and theoretical approaches to deep learning because the moment you start trying to impose axioms and

assumptions onto things and they do not perfectly track the underlying reality. Your theory is break. But I'm very hesitant to say anything's impossible. And I think there's far, far more to learn than we have.
SPEAKER: Tim Scarfe
Fantastic. Now PA says Question for Neil How does he see interpretability playing a role in security, not alignment. For example, crafting more exotic jailbreaks? And he says to tell you to blink twice if you can't answer due to an NDA.
SPEAKER: Neel Nanda
Yes, sorry. Jokes aside, what was the question? Um, so leaving aside things like alignment, do I think interpretability could be used to guard against more exotic jailbreaks and things like that? Um, and yeah, so I sure hope so. Um, I consider this kind of thing a more like a cute side effect of the process of getting better at interpretability. Like if we actually understand the system circuitry, we should be able to use this to design better adversarial examples to debug why certain jailbreaks work in ways that we wouldn't have expected. Um, and these are all going to be side effects of getting better at this stuff but currently not very good at it. So I think we're probably good enough at it that we could find some various ad hoc adversarial examples. Um, like one of my all time favorite adversarial examples is from the multimodal neurons and artificial neural networks paper where the signing this model called clip. Clip takes text and images and checks whether the text is a good caption and they find that clip is really good at reading text. And when people try to use clip as a classifier like kind of doing image net doesn't match apple or truck better, they find that if you take an apple and you put a sticker on it with the word iPod written on it, then the clip really thinks it's an iPod rather than an apple. And there's all kind of sketchy things about this. But this is just like a beautiful ideal to me of what interpretability inspired creative ways to break models looks like.
SPEAKER: Tim Scarfe
Now finally, Jumbotron Ian He says, Oh heck yeah. I'm glad to see that you brought this guy on. I've been interested in his work ever since you shared his blog. Now the question off the top of Ian's head is how does your theory, Neal, of chasing phase changes to create Grokking have any crossover or links with power law scaling techniques like in the, you know, scaling laws, paper beyond scaling laws, beating power law scaling via data pruning?
SPEAKER: Neel Nanda
Yeah, that is. Hmm. So we're going to get into this much more later in the podcast. But a very high level. I would say that Grokking is in many ways kind of an illusion as we'll get to later. And in one notable thing about it is grokking is a overlap between a phase transition where the model goes from cannot generalize to can generalize fairly suddenly and the phenomena where it's faster to memorize than to generalize. And these two things on top of each other give you this sudden memorization and failure to generalize followed by a sudden convergence later on. But the interesting thing here is the phase transition that's a much more robust result while Grokking is if you screw around with hyperparameters enough you get it to Grok. But it's very delicate and a little bit of an illusion. And this is a great paper from Eric Michaud and Max Tegmark's lab showing that well, providing a conceptual argument and some limited empirical evidence for the hypothesis that the reason we get these smooth scaling laws is that models are full of lots of phase transitions plausibly when they learn individual circuits. Though the paper does not explicitly show this and that the smooth scaling laws happen because there are just many, many phase transitions and if they follow a certain distribution you get beautiful smooth parallels. And to me this kind of thing is the main interesting link between broader macroscopic phenomena and these tiny things

though. And no, I also think Grokking is kind of overhyped and people significantly overestimate the degree to which it has deep insights for us about how networks work. And we think it's a really cute thing that gave me a really fun interpretability project and we learned a bit about signs of deep learning, but people often just assume it's like a really deep fact about models and yeah.
SPEAKER: Tim Scarfe
Okay, so we're now going to kick off with the actual Yes main podcast. So that was a nice little piece. Maybe we'll put that on the Patreons thingamajig. By the way, there was something I didn't say in the woods, which is that Neil has an amazing YouTube channel. I've been glued to it all week actually. Some of them are admittedly quite technical, but even if you're not interested in mechanistic interpretability, Neil has an extremely soothing voice, second only to Sam Harris and I would recommend listening to him when you go to sleep because as you know, Neil's dulcet tones will melt the stress away quicker than a nun's first curry. Anyway, with that said, we started to talk about what is mechanistic interpretability. And first of all, I wanted to call out your ridiculously detailed and exquisite mechanistic interpretability explainer. Maybe you could just tell us about that quickly.
SPEAKER: Neel Nanda
Yes. So I wanted to try to write a glossary for some basic common terms and McInturff is like an appendix to a blog post. There are a lot of terms in McInturff. There are a lot of terms in McInturff, and I like writing and I'm very bad at brevity, so I've got kind of carried away and there's over 33,000 words, massive, massive exposition. But importantly, it is designed to be easily searchable. And McInturff is full of jargon and I'm sure I'll forget to explain everything that I'm saying. So I'd highly recommend just having it open in a tab As you listen to this and if you get lost, just look up terms in there. And yeah, it's both definitions, but it's also long tangents giving intuitions and context and related work and common misunderstandings. It was very fun to write.
SPEAKER: Tim Scarfe
So I think first of all, we should introduce this idea of circuits and features and also this idea of whether interpretation is even possible at all. You know, why Why do you have the intuition that it is possible?
SPEAKER: Neel Nanda
Yeah. So a couple of different takes here. Um, so the key yeah. So fundamentally neural networks are not incentivized to produce legible, interpretable things. They are a mound of linear algebra. There's this popular stochastic parrots view that they are literally a mass of statistical correlations meshed together with no underlying structure. Um, the reason I think there's any hope whatsoever on a theoretical basis is that ultimately they are made of linear algebra and they are being trained to perform some tasks. And my intuition is that for many tasks the way to perform well on them is to learn some actual algorithms and like actual structured processes that maybe from a certain perspective you could consider Reasoning and models have lots of constraints like they need to fit it into these matrices. They need to represent things using the attention mechanism and jealous and a transformer and there's all kind of properties of this structure that constrain the algorithms and processes that can be expressed. And these give us all kinds of hooks we can use to get in on some what's going on. So that's the theoretical argument. All theoretical arguments are bullshit unless you have empirics behind it. And we're going to talk a bunch throughout this podcast about the different bit of different preliminary results we have that make me feel like there's something here that can be understood. What I find particularly inspiring is this work did reverse engineering modular addition which I think we'll get to shortly. Um, but kind of also want to

emphasize that C-mac and Terp as a bet. There's this strong hypothesis that if we knew what we were doing we'd be able to take GPT seven and fully understand it and decompile it to an enormous Python code file. And there's the weaker view that it is a mess and there's lots of illegible things, but we can find lots of structure and we can find structure for the important parts and make a bunch of progress. And then there's the yeah, we've cherry picked like ten things and the 11th is just going to completely fail and the field is going to get doomed and run out of steam in like a year and didn't really know I'm a scientist. I want to figure out I think it is worthy and dignified to make this bet. But I would be lying if I said I am 100% confident McInturff will work. Models are fundamentally understandable. We will succeed. Let's go try.
SPEAKER: Tim Scarfe
Well, on that note, how does it mean? We interviewed Christoph Molnar, who's one of the main classical interpretability guys. And I think everyone agrees in principle that you can't just look at the inputs and the outputs like a behaviorist. We need to understand why these models do what they do because sometimes they do the right things for the wrong reasons. So maybe first of all, without going too deep, I mean, could you just briefly contrast with, you know, classical interpretability?
SPEAKER: Neel Nanda
Yeah. So. So there's a couple of. Okay. So first off, I think it's very easy to get into kind of nonsense gatekeeping because there's both the cultural meetup community centered around. Chris Ola Not that much in academia though, some in academia and there's the academic fields of mechanistic interpretability, right? So there's lots of people doing work would consider mechanistic interpretability even if they don't engage much with the community or during it exists. For example, a friend of mine is Atticus Geiger. He's doing great work in Stanford at Stanford on causal abstractions, I believe discovered about a month ago that the mechanistic community actually existed. And I don't know, I don't like gatekeeping. Um, and there's lots of work that's kind of relevant but maybe not quite. McInturff Under a strict definition with those with that hedging out of the way, um, a couple of key principles. The first is inputs and outputs are not sufficient. And I think even within interpretability this is not a like uncontroversial claim. There's all kinds of things that are saliency maps attributing things to different bits of the inputs. There are things of the form train an extra head to output an explanation or just ask the model to output an explanation of why it does what it does. And I think that if we want something that could actually work for human level systems or even the frontier systems we have today, this is just not good enough. Particularly evocative example to me is in the Gpt4 System card, the Alignment Research Center and an organization they were getting to help audit and Red Team for had it try to help a TaskRabbit worker fill out a Captcha for it. The TaskRabbit worker was like Why do you need this? Are you a robot or something? GPT four on an internal scratch pad wrote out A I must not reveal that I am a robot. Um it then said Oh no, I've got a visual impairment and the transport worker did the Captcha and like this isn't some deep, sophisticated intentional deception, but it's very much like, well I don't trust the inputs and outputs of these models. Another really cute example is this paper from Miles Turpin that just came out about limitations of chain of thought where so chain of thought. You ask the model to explain why it does something. They were giving it multiple choice questions and asking it to explain its answer and then give the answer and they did five shot ish like here's five examples. Answer this question and then it modeled as well. And then they gave it something where all of the answers in the prompt are a correctly a, they just set it up. So the answer is A the model decides that it should output A um, but the model comes up with a

false chain of thought reasoning that gets it to the point where it says A is the right answer. And I don't know. Some people are trying to use chain of thought as an interpretability method and I think we need to move beyond this and engage with the internal mechanisms so that 0.10.2 is ambition. I believe that ambitious interpretability is possible or at least that if it's not possible, that striving for it will get us to interesting places. These models have legible algorithms. I want to try to reverse engineer them. Um, a third difference is engaging with the actual mechanisms and computation and algorithms learned. There's also work on things like analyzing features of a model, probing individual neurons and take this as very relevant to Macintosh. But I want to make sure we aren't just looking at what's inside the model but also trying to understand how it computes features from earlier features. What applying causal interventions to understand the actual mechanisms. Making sure we're not just doing correlational things like probing. And yeah, fourth is maybe a more meta principle of favoring depth over breadth, a kind of key underlying belief of a lot of my philosophy of interpretability is that it is so, so easy to trick yourself. There's all kinds of papers about the interpretability illusion impossibility, theorems for feature attribution methods. Various many ways that attempts to do interpretability have led to people confusing themselves or coming to erroneous conclusions. I think that if. But I also think that I want to be in a world where we can actually have scalable, ambitious approaches to interpretability that actually work for frontier systems but feel like we don't know what we're doing. And so my vision of mcinturf is start small, start where things where we can really rigorously understand what's going on slowly build our way up and like build a foundation of the field of interpretability where we genuinely understand rigorously what is going on and use this foundation to be more ambitious, to try to build real principled techniques, to be willing to relax the rigor, to be able to go further and see how far we can get And people and this means I'm happy with things like let's analyze an individual model and a understand a small family of features and a lot of detail rather than lots of stuff kind of jangly. There's a lot of stuff in summary, having an ambitious vision, not just looking at inputs and outputs actually try to engage with internal mechanisms and favoring depth over breadth. But I want to avoid gatekeeping, as I said, indeed, indeed.

SPEAKER: Tim Scarfe

What would interpretability look like in a world full of GPT four models and beyond? I mean, presumably you actually think that they're competent enough to deceive us and manipulate the inputs.

SPEAKER: Neel Nanda

I definitely want to clarify that when I say deception or manipulation here, I'm not making the strong claim that it's intentionally realized this for instrumental reasons as part of an overall goal I'm very happy with There was a prompt saying to deceive someone or it learned that in this context people often output things that are intended to convince someone and it just kind of does this as like as like a learned pattern of execution. Um, but yeah, my vision of what interpretability would look like is we take some big foundation model like the GPT four base model or the fine tuned GPT four that's being used as a base for everything else. We make as much progress as we can understanding the internal circuitry, both taking important parts of it and like important questions about it e.g. how does it model people it's interacting with? Does it have any notion that it is a machine learning system and like what would this even mean? And being willing to do pretty labor intensive things on that. Having a family of motifs and understood circuits we can automatically look for and very automated tools to make a lot of the labor intensive stuff as efficient as possible. Um, things like Openai's recent paper using GPT four to analyze GPT two neurons for like a very cute proof of concept here

though it needs a lot of work before it could actually be applied and rigorously and at scale and yeah, um, taking this one big model, trying to understand it as much as we can, one family of techniques we're going to get to is kind of causal abstractions and causal interventions which are very well suited to taking a model on a certain input or a certain family of inputs and understanding why it does what it does. There is a much more narrow and thus more tractable question than like what is GPT four? Um, and yeah, doing something like if there's a high profile failure, being able to debug it and really understand the internal circuitry behind that or yeah and know I have a bunch of other random thoughts. Um, one reason I'm emphasizing the focus on the big base model is I think a common critique is this stuff doesn't generalize between models or it's really labor intensive. But we live in a world where there is just like one big foundation model used in a ton of different use cases. Probably the circuitry doesn't change that much when you give it a prompt or you fine tune it a bit. And I think having getting a deep understanding of a single model is kind of plausibly possible.
SPEAKER: Tim Scarfe
But um, do you think it doesn't change that much?
SPEAKER: Neel Nanda
Uh, so no one's really checked. This is just true of so many things in interpretability. It's like, well, you know, my, my intuition is that when you fine tune a model, most of what is going on. Is that your. Arranging the internal circuitry so you fine tune Wikipedia, you upweight the factual recall circuitry, you flesh it out a bit, you downweight other stuff. And like I think this can explain a lot of improved performance, but then if you fine tune for much longer, you're basically just training the model and it will start to learn more circuitry, more features, more algorithms, more knowledge of the world. And yeah, but like no one's really checked. Um, and definitely the longer you fine tune it and the more you're using weird techniques like reinforcement learning from human feedback, the less I'm confident in this claim. Um, and yeah, if we discovered that every time you fine tune a model it will wildly change all of the internal circuitry. I'd be like somewhat more pessimistic about mankind up unless we can get very good at the automated parts which we might be able to get good at. Yeah, I very much think of the field as we're trying to do this highly ambitious thing. We're making a lot of progress, so I really wish we're making way more progress, way faster and you viewer could help. Um, but I don't know where the difficulty bar is for being useful or the difficulty bar is for being like incredibly ambitiously useful and it's plausible. We're already at the point where machines up can do real useful things no one else can or no other techniques can. Yeah, it's plausible. We'll take like five years to get to that point. I don't really know.
SPEAKER: Tim Scarfe
So I wanted to talk about this concept of Neats and Scruffies. So there been two divisions in AI research going all the way back to the very, very beginning. And you've said that sometimes understanding specific circuits can teach us universal things about models which bear on important questions. So this reminds me of this dichotomy between the Neats and the Scruffies. Now you seem like a neat to me, neat to someone who is quite puritanical and also it's related to universalism. So this idea that there are simple underlying principles that explain an awful lot of things rather than wanting to accept the gnarly kind of reality that everything's so bloody complicated, um, where do you fall on that?
SPEAKER: Neel Nanda
So I definitely would not say, Okay, so there's so there's two separate things here. There's like, what's my esthetic? Well, I want things to be neat. I want them to be beautiful. I want it to be mathematical. I want them to be elegant. And then there's what do I do in practice and

what do I believe is true about networks? Well, I think there is a lot more structure than most than many people think, but I also do not think they are just some beautiful, purely algorithmic thing that we could uncover if we just knew the right tools and like maybe they are we fucking great if they were. Um. But I expect they're messy and cursed but with some deep structure and patterns and how much traction we can get on the weird scruffiness is like somewhat unclear to me. I think we can make a lot more progress than we have, but we might eventually hit a wall.
SPEAKER: Tim Scarfe
But you were saying something quite interesting when we drove over, which is I mean my my friend Waleed Subir, he's a linguist and he's a platonist He thinks that there are these universal cognitive priors and there's a there's a hierarchy of them and and the complexity collapses. And he thinks that language models have somehow acquired these cognitive priors. And if we did some kind of symbolic decomposition, you know, we would all just kind of like pack itself into this beautiful hierarchy. And you were saying that there are Gabor filters and there are all these different circuits and they have motifs, they have categories, They have flavors, for want of a better word. Are you are you optimistic that something like this could happen?
SPEAKER: Neel Nanda
Yeah. So I'm. So one interesting one interesting point here is often interpretability is fairly different for different modalities and different architectures. A lot of the early work was done on convolutional networks and image classifiers. The field very much nowadays focuses on transformer language models and I think there's lots of structure to how transformers implement algorithms. Transformers cannot be recursive, but they're incredibly parallelized. Transformers have this mechanism of attention that tells them how to move information between positions and there's lots of algorithms and circuitry that can be expressed like this and lots of stuff that's really weird to express. And I think that this constrains them in a way that creates lots of interesting structure that can be understood and patterns that can be understood. And is this inherently true of intelligence? Who knows? But a lot of my optimism for structures within networks is more like that. But I try to think about structure more from a biologists perspective than a mathematicians or like philosophers perspective. Though I am a pure mathematician and I know nothing about biology, so if anyone's listening to this know software biology and thinks I'm talking bullshit, please email. Um, so if you look at evolutionary biology model organisms have all of this common shared structure like most things have bones. We have cell nuclei. Um, the hands of mammals tend to be surprisingly similar but like kind of weird and changed in various ways and I don't know, um. I don't think these are like hard rules. Most of them have weird exceptions and obviously a lot of this is due to the shared evolutionary history and is not just inherent to the substrate. If you have proteins though, the fact that you often train these models on similar data and similar ways and they have the same architecture that constrains them to different kinds of algorithms makes me optimistic. There's a biologists level of structure.
SPEAKER: Tim Scarfe
Now you said something interesting, which is that transformers can't be used in a recursive way. Now we're just touching this very quickly because we've spoken about this a million times on different episodes. But you know, there's the Chomsky hierarchy and he had this notion of a recursively enumerable language and these different models, computational models and the Chomsky hierarchy. It's not only about being able to produce a language which exists in a certain set. It's also the ability to recognize that the language belongs in a certain set and transformers are quite low down on that hierarchy because they're called

recurrently not recursively. But I just wondered if you had any just, you know, prima facie if you had any views on that.
SPEAKER: Neel Nanda
Yeah. So I'm not a linguist. I'm not particularly familiar with the Chomsky hierarchy. Um, I do think it's surprising how well Transformers work and I have a general skepticism of any theoretical hierarchy. Like I don't know if you think there's some beautiful structure of algorithms and stuff that's low down is totally doomed and then GPT four happens. I think your framework is wrong rather than transformers are wrong. Just massive stack of matrices plus a massive pile of data. Gives shockingly effective systems and theoretical frameworks just often break when they make contact with reality.
SPEAKER: Tim Scarfe
Well, that's certainly true. I mean, there's a famous expression that all grammars leak, but I had rather I don't know, I guess a similar conclusion to you, which is that if anything, it teaches us how sclerotic and predictable language is and we don't actually need to have access to this infinite space or even exponentially large space. Most language use and most phenomena that we need perhaps for intelligence is surprisingly small and current models can can work just Well, why don't we move on to your grokking work? So grokking is this sudden generalization that you know happens much later in training after?
SPEAKER: Neel Nanda
If I can add a brief clarification. Oh yes, of course. So people often call Grokking sudden generalization.
SPEAKER: Tim Scarfe
My apologies.
SPEAKER: Neel Nanda
Go on. Sudden generalization is a much more common phenomenon than grokking. It can just generally look like things like I know the model trying to learn a task. It's kind of bad at it and then it suddenly gets good at it. And I prefer to call this a phase transition, right? Grokking is the specific thing where the model initially memorizes and does not generalize and then there's a sudden phase transition in the like test loss, the generalization ability which creates a convergence after an initial divergence between train and test. And this is like a much, much more specific phenomena than sudden generalization in general.
SPEAKER: Tim Scarfe
Okay. Well, so you've spoken.

About.
SPEAKER: Tim Scarfe
You've spoken about three distinct phases of training underlying Grokking. So why don't we go through them one by one?
SPEAKER: Neel Nanda
Yeah. So the context of this project, this was a paper called Progress Measures for Grokking via mechanistic interpretability that I've recently presented at at Prentice presented on at Iclear. The Yeah. So we were studying a one layer transformer. We trained to do modular addition and it dropped modular addition and the first thing we did was reverse engineer the algorithm behind how the model worked, which we may get into in a bit more detail, but at a very high level modular addition is equivalent to composing rotations around the unit circle Composition adds the angles Circle gives you modularity. You can represent this by Trig functions and do composition with Trig identities and element wise multiplication. And we reverse engineered exactly how the model did this. And then this mechanistic understanding

was really important for understanding what was up with Grokking because the weird thing behind Grokking is that it's not that the model memorizes all that the model eventually generalizes. The surprising thing is that it first memorizes and then changes its mind and generalizes later and. Generalization. Memorization are two very different algorithms that both do very well on the training data and only by understanding the mechanism will be able to disentangle them. And this meant we could look during training how much of the model's performance came from memorization and how much came from generalization. And we found these three distinct phases. There was memorization, the first very short phase. It gets phenomenally good trainloss. It got to about three minus seven, which is an absolutely insane log loss and much, much worse than random on test because memorization is very far from uniform and generalizes extremely badly. And then there was this long seeming plateau. We call this phase circuit formation because it turns out that rather than just continuing to memorize for a while and doing a random walk through model space until it eventually gets lucky, the model is systematically transitioning from memorization to generalization. And you can see that it's trained performance gets worse and worse when you only let it memorize. And then so why is Tesla still bad? Tesla's is bad because memorization generalizes terribly. And when the model is like, I don't know, two thirds memorizing one third generalizing, this still does terribly and it's only when the model gets so good at the Trig based generalizing algorithm that it no longer needs the memorization parameters and cleans them up that we see grokking. And this happens fairly suddenly. But the if you we have this metric called restricted loss where we explicitly clean up the memorization for the model and look at how well it generalizes and we see that restricted loss drops noticeably before test loss drops showing that the drop is driven by cleaning up the noise. And this is striking because A, I had no idea it was even possible for a model to transition between two good solutions maintaining equivalent performance throughout. B There was this real mystery of deep learning that many people tried to answer and mechanistic understanding was genuinely useful for answering it. And Grokking was an illusion. It was not sudden generalization. It was gradual generalization followed by sudden cleanup and test loss and test accuracy were just too coarse a metric to tell the difference. But we were able to design these hidden progress measures using our mechanistic understanding that made everything clear. And we also have all kinds of pretty animations of qualitatively watching the circuits develop over training and it's very pretty.
SPEAKER: Tim Scarfe
So a few things. I mean, first of all, just going back to first principles, the biggest problem in machine learning is this concept called overfitting. And we trained the model on a training set and there's this horrible phenomenon called the shortcut rule, which is that the model will take the path of least resistance. And when you're training it, it only really knows about the training set. And of course we can test it on a different set afterwards, which we've held out. And just because of the way that we've structured the model, it may by hook or by crook generalize to the test set. But the interesting thing is that generalization isn't a binary. There's a whole spectrum of generalization, so it starts with the training set and then we have the test set and then like, you know, the ideal is out of domain generalization but would go a step further. There's also algorithmic generalization, which is this notion that as I understand it, neural networks, if you if you model the function Y equals X squared, it will only ever be able to learn the values of that function inside the training support. So presumably you're talking about the ideal form of generalization being not as good as algorithmic generalization or do you think it could go all the way?
SPEAKER: Neel Nanda

So I think one thing which is very important to track is what's the domain you're talking about is of which it's even possible to generalize. So I generally think about models that have discrete inputs rather than continuous inputs because basically no neural network is going to be capable of dealing with like unbounded range, continuous inputs. Um, in modular addition, there were just two one hot encoded inputs between 0 and 113, which is the modular I used. Yeah, the model has a fixed modular. It's not doing modular addition in general and there's just like 12,000 inputs and it learns to do all of them. And in I don't know behaviorally you can't even tell the difference between the model memorizes everything and the model learns some true algorithm though with the more cognitivist mechanistic approach I can just look at it and say Yep, that's an algorithm. It's great, not a stochastic parrot conclusively disprove that hypothesis. Um, and yeah. I think that for language models it's more interesting because I know gpt2 it's got a thousand tokens, 50,000 vocab that's like 50,000 to the power of a thousand possible inputs. And there's a surprising amount of interesting algorithmic generalization. Um, we're going to talk later about induction heads, which is this circuit language models learn to detect and continue repeated text like if given the word Neil, you want to know what comes next? Unfortunately, Nanda is not that high on the list yet. Um, but if Neil Nanda has come up like five times before in the text, Nanda is pretty likely to come next. And, um, this transfers to if you give the model just random tokens with some repetition, the model can predict the repeated random tokens because the induction heads are just a real algorithm and the space of possible repeated random tokens is like enormous. It's like in some sense much larger than the space of possible language. And is this algorithmic generalization? Don't really know. It depends on your perspective.
SPEAKER: Tim Scarfe
Let's bring in this paper by Bilal Chugtai. So it was called a toy Model of Universality. Reverse engineering How Neural Networks Learn Group Operations. And you supervised that paper and he was asking the question of whether neural networks learn universal solutions or these idiosyncratic ones. And he said he found inherent randomness. But models could consistently learn group composition via an interpretable representation theory. So can you give us a quick tour de force of that work?
SPEAKER: Neel Nanda
Yeah. Maybe I should detour back to my grokking work and just explain the algorithm we found there and how we know it's the real algorithm. Yeah, sure. It's a good foundation for this paper. Sure. Sure. Yeah. So we found this thing we call the Fourier multiplication algorithm. The very high level it composes rotations. You can actually look at how the different bits of the model implement the algorithm and often just read this off. So the embedding. So just a lookup table mapping the one hot encoded inputs to these Trig terms sines and cosines of different frequencies. You can just read this off the embedding weights. Note People often think that learning sine and cosine is hard. It's actually very easy because you only need it on 113 different data points. So just a lookup table. The model then uses the attention and MLPs to do this composition, to do the multiplication with Trig identities to get the like composed rotation. The A plus B terms. And here we can just read off the neurons that they have learned these terms and that they were not there beforehand. The model is using its non-linearities and interesting ways to do this. It's also incredibly cursed because Relu is are not designed to multiply two different inputs, but it turns out they can if you have enough of them and a sufficiently cursed. Um, and yeah we can just read this off the neurons. Also if you just plot anything inside the model, it's beautiful and it's so periodic and I love it.
SPEAKER: Tim Scarfe

Um, could I touch on that though? Because you said you don't need to know the sine function because you can just measure it, memorize it within an interval. Is that is that I don't know. How does that break down Because it's it's discretizing it and it's kind of assuming that it has the same behavior in different intervals.
SPEAKER: Neel Nanda
So I think a key thing here is that you are solving modular addition on discrete one hot encoded inputs rather than for arbitrary continuous inputs. Arbitrary continuous inputs is way harder. And so you it's not even on an interval, it's just learning snapshot, it's just learning like single points on the sine and cosine curves. And I don't know, there's this family of maths about studying periodic functions with different kinds of Fourier transforms and this is all discussing discrete Fourier transforms which are just a reasonable way of looking at periodic sequences of length. N And that's how I recommend thinking about this one. Um, it's kind of like just quite different from a model that's trying to learn the true sine and cosine function. Um, and yeah, um, the model then needs to convert the composed rotation back to the actual answer, which is an even more galaxy brained operation that you can read off from the weights. So you've got terms of the form cos A plus B the model has some weights mapping to each output C and it uses further Trig identities to get terms of the form cos A plus B minus C times some frequency and where A and B are the two inputs, C is the output and you then use the softmax as an argmax to like extract the c that maximizes this and because cos is maximized at zero, this is maximized at C equals A plus B and if you choose the frequency right this gets you mod n and you can just read this off the model weights. It's great. And then finally you can verify you've understood it correctly because if you ablate everything that our algorithm says should not matter, performance improves. While if you oblate any of the bits our algorithm says should matter performance tanks.

Okay.
SPEAKER: Tim Scarfe
Could you give me some intuition though? So we start off in the memorization phase because guess you can think of a neural network as doing many different things in a very complicated way and there's some kind of change in the balance during training. So it does the easy thing first and then it gradually learns how to generalize. And in this particular case, how does that thing because we're using stochastic gradient descent. So we're moving all of these weights around and the inductive prior is also very important and we'll come to that I think after we've spoken about Bilal's paper. But how does that happen? Gradually in really simple terms.
SPEAKER: Neel Nanda
Is the question then kind of it ends up at this discrete algorithm, but it does so by continuous steps. How does that work?
SPEAKER: Tim Scarfe
Well, I think the thing that surprised a lot of people about Grokking is this I mean, Grokking the clue's in the name, so it's gone from memorization and then we're using stochastic gradient descent and you would think that it's gotten stuck in some kind of local minimum and you're training and you're training and you're training and then there's a spark. Something happens and then the you get these new modes kind of like emerging in the network. I'm not sure if emerging is the right term and it happens gradually and it happens after a long time.
SPEAKER: Neel Nanda

Yeah. So there's a couple of things here that's pretty easy to misunderstand. The first is that the first is that I think it's pretty hard for a model to ever get stuck because I know this model had about 200,000 parameters. Modern ones have billions. It's just moving in a very high dimensional space and you can get stuck on 150,000 dimensions, but you've got 50,000 to play with and especially for a fairly under parametrized model for a fairly overparametrized model like this one for a fairly simple task. They're just like so much room to move around. Um, another common misunderstanding of grokking is people say it's memorize, it's got zero loss. So why does it need to learn to misunderstandings here first zero loss is impossible unless you have bullshit floating point errors because it's log. It's like the average correct log prop log of anything can never get to the log, will never quite get to zero because of just how softmax works and you need to have an infinite logit for that to happen. Um though one cute thing in an appendix to our paper is that float 32 cannot represent log probs less than 1.19 minus seven which leads to bizarre loss spikes sometimes unless you use float 64. Anyway. Yeah. The second is regularization. If you don't have any kind of regularization, the model will just continue to memorize. We use weight decay Dropout also works and so the model, the kind of core tension behind Grokking is there's some feature of the loss landscape that makes it easier to get to memorization. You can memorize faster while generalization is somehow hard to get to and much more gradual. So the model memorizes first, but it ultimately prefers to generalize. But it's only a mild preference. And the reason for this is we cherry pick the amount of data where it's a mild preference because there's too little. It will just always memorize. If there's too much, it will immediately generalize because you know, grokking is a little bit. Sheeting and yeah, you then use this and because the models initially memorized but it wants to generalize it can follow it memorizes until the desire to memorize more balances with the desire to have smaller weights. But both of these reinforce the drive to generalize because both because that makes both of them happier. And so the model very slowly interpolates very, very slightly improving test test loss, very slightly improving train loss until it eventually gets there. And has this acceleration at the end. This phase transition and clean up which leads to these seemingly sudden grokking behavior.

Okay.
SPEAKER: Tim Scarfe
And when you were talking about the it wants the weights to be smaller so that's weight decay and it's like an inductive bias essentially to tell the model to reduce its complexity, which is a pressure to generalize. But if it wasn't for that, then that wouldn't happen.
SPEAKER: Neel Nanda
So in the experiments I ran, if you don't have weight decay, it will just keep memorizing infinitely far because when you get perfect accuracy, if you double all your logits you just get more confident in the right answer. And so it just keeps scaling up. I was using full batch training because it's such a tiny problem. This made things smoother and easier. I've heard some anecdata that sometimes you can get it to work if you just have mini batch stochastic gradient descent, but I haven't looked into that particularly hard.

Interesting.
SPEAKER: Neel Nanda
There are some hypotheses that Stochasticity acts as an implicit regularizer because it adds noise. I don't really know.
SPEAKER: Tim Scarfe

So let's go back to Bilal's paper then. So this paper a toy model of universality, Reverse engineering, how neural networks learn Group operations. Can you give us an elevator pitch?

Yeah.
SPEAKER: Neel Nanda
So an observation that actually first discovered at a party in the Bay Area from a guy called Sam Marks is that the modular addition algorithm we found is actually a representation theory algorithm. So group representations are kind of collections of symmetries of some geometric objects that correspond to the group. Modular addition is the cyclic group and rotations of the of the regular n gon are the representations of the cyclic group and this corresponds to the rotation about the unit circle that compose that we found. But it turns out you can just make this work for arbitrary groups. You replace the two rotations with just two representations, you compose them and the model and it turns out the cause A plus B minus C thing is this math jargon called the character you don't lean to on sign any of that. But it's very cute if like me, you have a pure math degree And for example, if you have the group of permutations of five elements, the 120 different ways to rearrange five objects. One example of representations of this are rotations and reflections of the four dimensional tetrahedra. And if you train a one hidden layer MLP to grok this and look inside, you can just see these rotations that it's learned. It's gorgeous. And so the first half of that paper was just showing that the algorithm worked, showing that this was actually learned in practice. Then the interest then the more interesting bit was this focus on universality. So universality is this hypothesis that models have some intrinsic solutions to a problem that many different models will converge on, at least given similar data and similar architectures e.g. in image models. Models will learn specific neurons that detect curves and different models and different datasets seem to learn this similar thing. And here this was interesting because groups have a finite set of irreducible representations. Math's Theorem You can enumerate these, there aren't that many of them. And for groups that are not modular addition, these are qualitatively different. Like some of them act on a four dimensional object like the tetrahedron and some of them act on like 5 or 6 objects. Naively, some of them are simpler than others, but they're definitely different. And so what we did is we asked ourselves the question. Which one does the model learn? And we found that as you even if you just vary the random seed, the model will randomly choose a subset of these each time to learn. And there's some structure like it tends to learn some of them more often than others. This a little bit maps to our intuitive notion of simplicity, but not that much. One of the updates I made in the paper is that simplicity is a really cursed concept. Don't understand very well where I don't know if you have rotations of a four dimensional object that seems simpler, but maybe the 60 object takes more dimensions but has better loss per unit weight norm which is simpler. I don't know. Um, but yeah. Anyway, we found that each run the model learns some combination of these circuits for the different representations. It's like normally more than one. The exact number varies and which ones it learns is seemingly random each time, which suggests that all toy models lie to you obviously. But if we're trying to reason about real networks, looking at this work might suggest the explanation, the hypothesis that there are if there are multiple ways to implement a circuit, which in practice then normally are models may learn different ones of them, kind of a fairly random reasons and that fully understanding one model will not perfectly transfer to another model. And I think there's like loads of really interesting open questions here. Like I know people have done various work understanding different kinds of specific circuits and models like the Interoperability and the

wild paper we'll get to later. What does this look like in other models? Um, often there's multiple ways to implement a circuit. Can you disentangle the two? Are all models learn both or do some models learn one, some learn the other? I don't really know.
SPEAKER: Tim Scarfe
So a couple of questions. I mean, um, first of all, this is leading towards this idea that we were speaking about before, which is that, um, even in different networks, slightly different problems or variations on the same problem, it could learn these algorithmic primitives. Now the first observation here is that the, um, the inductive biases of, of, of the network differ massively. Right. So to what extent do the inductive biases affect these primitives which are learned?
SPEAKER: Neel Nanda
Oh, so much.

They do.
SPEAKER: Neel Nanda
So.

Well.
SPEAKER: Tim Scarfe
Could I frame the question a little bit because this reminds me a lot of um, the geometric deep learning blueprint from, uh, Peter and Michael Bronstein and all those guys. And they were coming at this from exactly the same direction as you. They said there's a representation of a domain which is basically a symmetry group and you can do all of these different transformations. And and as long as they fall in different positions and the underlying domain so they respect the structure, then it works. But all of those, um, all of those symmetries are effectively coded into the inductive prior. So for example, if a CNN works on this gridded 2D manifold and it explicitly um models translational equivariance and local connectivity and weight sharing and so on. So I guess what I'm saying is like you're talking about this four dimensional tetrahedron and that isn't explicitly modeled in an MLP. So so how are you even recognizing that it's learning those symmetries? How are you even probing it? Maybe we should start with that.
SPEAKER: Neel Nanda
Uh, so I guess thing one models are just smarter than you and models can do a lot of weird stuff. Uh, I feel like the story of deep learning is people initially thought they needed to spoon feed these models the right inductive biases over the data. Um, and we've gradually realized, oh, wait, no, no, this is fine. The models can figure it out. For example, so early on image models were convolutional networks. You tell it, the key information is nearby and if you translate the image, it doesn't matter. And now everyone uses transformers including for images and transformers. Replace the convolutional mechanism with attention where you're now saying, okay, one sixth of your parameters are dedicated to figuring out where to move Information between positions sometimes will be a convolution and sometimes models do learn convolutions, but often it won't be. And we want you and you can now spend the parameters to figure this out. And I'm not very familiar with the with the geometric deep learning literature, but I generally am just kind of like models can figure it out. The way we figured out that this was what's going on is kind of analogous to what we did in the modular addition case where we just look at the embedding matrix and just read off the learned sine and cosine terms. Here we said, okay, the rotations of the 4D tetrahedron are these like 4x4 matrices. You can flatten this to a 16 dimensional vector. Let's probe for that linearly. And this

kind of works and you can probe for the different representations and basically see what's going on.
SPEAKER: Tim Scarfe
Okay. But I think that the thrust of the geometric deep learning stuff or any inductive prior comes back to the bias variance trade off and the curse of dimensionality. So no one saying of course an MLP the if you look at the function space that it can approximate, it's exponentially larger than that of a CNN. So so it was always about sample efficiency. So yeah an MLP can learn anything, but we would never be able to train it for most problems.
SPEAKER: Neel Nanda
Yeah. So I guess I maybe want to avoid going too deeply into this because I think the module edition problem and the group problem is just a very weird problem. There's an algorithm that it's fairly natural for a model to learn with literally a single non-linear step of multiplication of like the matrix multiply. Um, one very cute result from Bilal's paper is that the model can implement two 4x4 matrix multipliers with a single Relu layer, which is very cute. Um, but yeah, there's like a fairly natural algorithm to implement. It's a certain yeah. Another useful intuition is that the more data you have, the more complex memorization gets. While generalization is exactly as complex at each point. And yeah, um, so there's kind of always going to be a crossover point if you have enough data where it is simpler to learn the circuit that generalizes. Um, and I don't know, I'm hesitant to draw too much from toy models about the real problem. I guess one two final points I'd want to just leave on this section. The first is I just want to reemphasize I did not do the toy model of universality paper. I was supervising a mentee, Bela Chugtai, who did it, who did a fantastic job. So thanks, Bela, if you're listening. Um, secondly, um, for the model prediction case, I had no idea this album was going to be there. When I went in, I just poked around, noticed the weird periodicity, realized it was using I should apply Fourier transforms and then the whole problem kind of fell together And to me that like to me the real takeaway of this paper is like don't give a fuck about Grokking It is genuinely possible to understand what is going on in a model. You don't need to know what's going on in advance to discover this and there is beautiful non-trivial structure that can be understood and who knows if this will happen in like actual full models. But to me this is much more compelling than if we had nothing at all.
SPEAKER: Tim Scarfe
Beautiful. Okay. And just before we move off the section, Bilhaud had a beautiful Twitter thread actually, and he was talking about the the potential for what he called a periodic table of universal circuits. Um, and I actually think that's a really cool idea, so that would be amazing if that would work out. But he also brought up the lottery ticket hypothesis and I've interviewed Jonathan Frankel and the idea there is that, um, some of this information might actually be encoded and understandable at initialization before you even start training. And apparently you folks had found weak evidence for this in at least one group.
SPEAKER: Neel Nanda
Uh, all right. So a couple of things there. Um, so this idea of a periodic table of circuits I believe is originated in this post called Circuit Zoom in from Chris Oehler. Um, we probably cannot claim credit though. It's it's a beautifully evocative term. Yeah. Yeah. The story of basically everything in McInturff is. Yeah, there's this kristeller paper from like two years ago that has it somewhere inside. Um, Anthropic recently put out this beautiful blog post called Interpretability Dreams about their vision for the field of mechanistic interpretability and the kind of subtext they kept just quoting bits of old papers being like, So we already said this, but let's now like summarize it better and be clear about how this fits into our overall picture anyway. So yeah, the idea of the periodic table is maybe there is just some finite list of ways

a thing can be implemented naturally in a massive stack of matrices that we can enumerate by studying one or maybe several networks, understand them and then compile all of this into something beautiful and which is kind of what we found in the representations case though here. It was nice because there were. Genuinely a finite set that we could fully enumerate regarding the lottery ticket stuff. I think this was a random observation I had on the modular addition case. Partially inspired by a result from Eric Michaud at MIT who was involved in some other papers on Grokking. And so what we found is that at the end of training there are these directions in the weights that represent like the sign in terms of frequency 14 pi over 113. And if you look at the embedding at the start and projecting onto these directions, it's like surprisingly circular. It's it's like the model has extracted those directions and my wildly unsubstantiated hypothesis for why models learn these algorithms and circuits at all is that there are some directions that if you delete everything else would like form this beautiful circuit. This is kind of a trivial statement about linear algebra for the most part and this underlying hidden circuit each bit reinforces each other systematically because they're useful. Well, everything else is kind of noise, so it gets kind of gradually decayed. And so over time this will give you the the circuit in a way that looks surprising and emergent. And this also can partially explain why phase transitions happen. Those are a really good post from Adam German and Bach Schlagers called on S-shaped curves which argue that if you've got something that's like the composition of multiple different weight matrices, let's just say two of them, the gradient on the first is proportional to how good the second is and vice versa. So the start they both grow very slowly, but then they'll reinforce each other and eventually cascade as they're optimizing on the problem in a way that looks kind of sudden and S-shaped. And so my understanding is the original lottery ticket hypothesis is kind of discrete. It's looking on the neuron level and it's learning masks over weights and over neurons. And I'm kind of discussing and in some sense much more trivial version where I'm not assuming there's some canonical basis of neurons. I'm saying, well, there's some directions in space that matter and if you delete all other directions, everything kind of works, which I think is a much more trivial statement though the space of possible neurons is enormous though I don't know one thing you want to be pretty careful of when discussing this stuff is how much the mask you learn is the computation. Since I know there's probably quite a lot of algorithms that can be cleverly expressed with a mask over a Gaussian normal matrix, but I don't know.

SPEAKER: Tim Scarfe

Part two How do machine learning models represent their thoughts? Now we're taught in machine Learning 101 that neural networks represent hypotheses which live on a geometric domain and inductive priors. Learn to generalize symmetries which exist on the underlying geometric domain and you're talking about them representing a space of algorithms which we're going to explore. Now one thing that I wanted to touch on is that they learn a mapping to extensional attributes, not intentional attributes. Intention spelt with an S and we'll come back to what I mean by that in a second. But I think it's quite popular for people to think of neural networks principally as a kind of hash table so or a locality sensitive hash table. And the generalization part comes from the representation mapping function which is on this embedded Hilbert space, which is the vector space of the attributes which then resolves a pointer to a static location on the underlying geometric domain. Now this can mimic an algorithm, especially when the inductive prior itself is increasingly algorithmic like a graph neural network for example, which behaves in a very similar way to a prototypical dynamic programing algorithm. There's some great work actually on algorithmic reasoning by Peter Velickovic, one of your colleagues now at DeepMind, but he showed in his algorithmic

reasoning work that Transformers can't perform certain graph algorithms. I think he gave Dijkstra as an example and he said it's because there's this aggregation function in a transformer which isn't in a GNN. So I just wondered if you could kind of like compare and contrast whether or not neural networks are performing algorithmic generalization and the differences between, let's say, Gans and Transformers.
SPEAKER: Neel Nanda
Yeah. So I'm not very familiar with Gans, so I'll probably avoid commenting on Dnns versus Transformers for fear of embarrassing myself. Um, in terms of the underlying thing. So I definitely think we have some pretty clear evidence at this point that models are doing some genuine algorithms. Um, and I think my model audition thing is a pretty clear proof of concept of this. I think that. Yeah. One Yeah, One thing which I think is interesting is I normally think of models as having what, what I'll call linear representations more so than geometric representations. Yeah. So one thing worth stressing is that I generally think of models as having linear representations more than geometric representations. So I think of an input to a model as having many different possible features where features are kind of a property of the input in an intentional sense. Um, but which is kind of a fuzzy and garbage definition. So I prefer the extensional definition of like an example of a feature is like this bit of an image contains a curve or this bit of an image corresponds to a car window or this is the final token in Eiffel Tower or this corresponds to a list variable in Python with at least four elements and all kinds of stuff like that. And I know this this scene is shaded blue because someone put the wrong filter on the camera. And yeah, I generally think of models as representing features as linear directions in space and each input is a linear combination of these directions and this is kind of the classic word two vec framing like the king minus man equals queen minus woman thing where you can kind of think of this as there being a gender direction and there being a royalty direction. And these are like the right units of analysis rather than king queen man, women being the right units of analysis. But where each of these is made up out of these underlying linear representations and this is a fairly different perspective to the geometric where are things in a manifold? How close are they together in Euclidean space? Because that's all that's all kind of a global statement about how close two things are where you're comparing all possible features. While I don't know, the Eiffel Tower and the Colosseum are close together in some conceptual space because they are both European landmarks, but they're also very different because France and Italy are fairly different countries in some sense and maybe they're different on a bunch of other features or one of them is two words, the other is one word which really matters in some ways. And Euclidean distance and geometry is it's a global summary statistic and all summary statistics Lie to you is another motto of mine, but in particular global ones I'm very skeptical of. And yeah, in general this how what is the structure of a model? Representations I think is like a really important question and in particular models are such high dimensional objects that you really want to be careful to distinguish between the two separate things of sorry, say that again. Um, models are such high dimensional objects that it's basically impossible to understand Gpt3 As a 200 billion dimensional vector, you need to be breaking it down into units of analysis that can vary independently and are independently meaningful. And the linear representation hypothesis is like a pretty load bearing part of how we think about this stuff because it is so because it allows you to break things down and it seems to be a true fact about how models do things. Though again, we don't have that much data because we never have enough data. It's really sad. And yeah.

Um.

SPEAKER: Tim Scarfe
Well let's contrast a little bit. So, so this linear representation hypothesis, this idea that the models break down inputs into many independently varying features and store them as directions in space, much like word two Vec and the the Go people. I mean like Fodor and Pylyshyn, they they brought out this famous critique of Connectionism in 1988 and their main argument was Systematicity and they were talking about intention versus extension and it might just be worth defining what I mean by that. So if I said the teacher of Socrates was Plato, the extension is Plato. The intention is everything. It's the teacher, it's Socrates. You know, if I said four plus five equals nine, nine is the extension four and plus and five is the intention. So. They were saying something very simple. They said in a neural network the intentional attributes get discarded and that's why the networks don't support what they call compositionality. Now, Compositionality is actually quite an abstract term because using vector algebra in these analogical reasoning tasks that you were just talking about so king and queen and so on, that's a form of compositionality. But they would say it's a poor cousin of Compositionality because it's only using, you know, the. The representation is in a is in a vector space and in a vector space you only have very basic primitive transformations. So you wouldn't be able to I mean, for example, you're talking about Paris earlier. You wouldn't do the kind of analogical reasoning they were talking about being able to downstream say were they in Paris? Is Paris in Europe? Of course it does happen in in this linear representation theory, but it happens in a very different way.
SPEAKER: Neel Nanda
Hmm. So I guess I'm not sure I fully followed that. Um, I mean, this might be a cheap gotcha, but a fact about Transformers is there's they have this central object called the residual stream, which I know in standard framing can be thought of as the thing that lives in the skip connections but not even is like the key thing about a transformer where each layer reads its input from the residual stream and adds its output back to the residual stream. And the residual stream is kind of this shared bandwidth and memory. And this means that nothing is ever thrown away unless the model explicitly is trying to do that or is just applying some gradual decay over time. So you know, if you've got an MLP layer that's saying I've got for I've got plus I've got five and I want to compute nine for five and plus is still there. I don't know if that's actually engage with your points and like I don't know if this matters but it's true.
SPEAKER: Tim Scarfe
Yeah. What you're saying is true. But I think the point is that those primitives are not actually representable in a neural network. So you're saying with this residual stream, all of the extensions that came previously also get passed up. So in a later layer you can refer to an extension. So the basically the answer of a computation that happened upstream. But what you can't refer to are the intentional attributes of that computation upstream.
SPEAKER: Neel Nanda
Why not like for is an input so you can refer to for because you could think of reading the input as a computation plus is another thing you read. Five is another thing you read like what is a thing that is not an output of a computation within this framework?
SPEAKER: Tim Scarfe
I might have to get back.

To you on that.
SPEAKER: Tim Scarfe

Oh, where's Keith Dugger when you need him? What would be a good example of that? Hmm. I mean, I guess it's it's about symbol manipulation as well. So these these things could actually be symbolic operations which can be composed and reused later. And you would appreciate that a neural network is only ever passing values. So, for example, if it did something which you could represent with a symbolic operation, if you wanted to use that again, I mean in an MLP, the reason why we use a CNN is because we want to represent the same thing in different places and an MLP would have to learn it. It doesn't support translational equivariance, so it would have to learn the same thing a million times. And it's the same thing with this symbolic compositional generalization that if it actually had this symbolic representation which it used once, it could use it everywhere, but now it has to relearn it everywhere.
SPEAKER: Neel Nanda
Mm hmm. Right. Like you could if if the model wants to know that Paris, the capital of France, it can spend some parameters on that. And for every other capital, it needs to separately spend parameters. And it can just have a general map country to capital operation.
SPEAKER: Tim Scarfe
Yeah, that's exactly right. I mean, let's let's use a simple example. So we use an MLP image classifier and I put a tennis ball in and it's it's in the bottom left of the visual field. Yeah. And and then I put it in the top right and nothing it's learned from the bottom left will be used.
SPEAKER: Neel Nanda
Mm hmm.
SPEAKER: Tim Scarfe
So it just it just feels like we're we're wasting the representational capacity just doing the same thing again and again and in a transformer. The only reason it does have that, um, recognition, you know, that that equivariance in respect of the position of a pattern is because of the transformer inductive prior presumably.
SPEAKER: Neel Nanda
Yes. Yeah. So it uses the same parameters at each position in the input sequence. So it should be able to do bottom left and top right properly though it does not necessarily have things like rotation built in. Um, I don't know. I feel like machine learning is full of these people who have all kinds of theoretical arguments and then they're like, This should be efficient, this should not work. And then GPT four laughs at them and I don't know. No theory. No theory is interesting in isolation unless it models reality. Well, and I don't know I haven't really engaged with this theory in the same way I haven't engaged with most deep learning theory because it just doesn't seem to meet my bar of Does this make real predictions about models? The maximal update parameterization paper from Greg Yang was actually a recent contradiction to this right of really interesting theory that makes real predictions about models that bear out and get you zero shot hyper parameter transfer. But like most things just don't do that.
SPEAKER: Tim Scarfe
Very interesting. Okay. Okay. Well, I think now is a beautiful opportunity to move over to Othello. Now, there was a recent paper called Do large language Models Learn World models or are they just Surface statistics by Kenneth Li? And he said that the recent increase in model and data size has brought about qualitatively new behaviors such as writing code or solving logic puzzles. Now he asked the question How do these models achieve this kind of performance? Do they merely memorize training data or are they picking up the rules of English grammar and grammar and the syntax of the C language? For

example? Are they building something akin to an internal world model, an understandable model of the process producing the sequences? And he said that some researchers argue that this is fundamentally impossible for models trained with guess the next word to learn the language meanings of language and their performance is merely surface statistics, which is to say a long list of correlations that do not reflect a causal model of the process generating the sequence. Now you said, Neal, that a major source of excitement about the original Othello paper was that it showed that predicting the next word spontaneously learned the underlying structure generating its data. And you said that the obvious inference is that a large language model trained to predict the next token may spontaneously model the world. What do you think?
SPEAKER: Neel Nanda
Uh, yes. So I should clarify that that paragraph was me modeling why other people were excited about the paper. Okay, but whatever. I can roll with this question, so.

And.
SPEAKER: Tim Scarfe
Maybe bring in your less wrong Yes.

As well. Yeah.
SPEAKER: Neel Nanda
Yes. So the Yeah. Thought Kansas paper was super interesting. The exact setup was they train so Othello is this chess and go like board game they took a dataset of random legal moves in Othello. They trained a model to predict the next move given a bunch of these transcripts. And then they probed the model and found that it had learned a model of the board state despite only ever being told to predict the next move. And so the way I would define world model is that there are some latent variables that generate the training data. Um, in this case what the state of the board is um, these change over time like over the sequence, but at least for a transformer which has a sequence and the model kind of has an internal representation of this at each point and they showed that you can probe for this and they showed that you can causally intervene on this and the model will make legal moves in the new board even if the board say is impossible to reach.
SPEAKER: Tim Scarfe
Point of order. Can you explain what you mean by probe just just so that the listeners know?
SPEAKER: Neel Nanda
Yes. So probing is this like old family of interpretability techniques? The idea is that you think a model has represented something like you give it a picture and you tell it to classify the image and you want to see if it's figured out that the picture is of a red thing versus a blue thing. Even though this isn't an explicit part of the output, you take some neuron or layer or just any internal vector of the model and you train some classifier to map that to like red or blue and you do something like a logistic regression to see if you can extract whether it's red or blue from that. And there's also interesting stuff about probing, but I should probably finish explaining the Othello paper first before I get into that tangent, please. So yeah, the like reason people are really excited about this paper was recently an oral at Iclr and generally got a lot of hype was that it was just you train something to predict the next token and it forms this rich emergent model of the world and forming a model of the world is actually incredibly expensive. They like each cell of the 64 cell. Othello board has three possible states, 3 to 64. It's quite a lot of information to represent. But the model did it and lots of people were like, Oh, clearly language models have won models. Um, my personal

interpretation of all this is that language models predict the next token, they learn effective algorithms for doing this within the constraints of what is natural to represent within transformer layers. And what this means is that if predicting the next token is made easier by having a model of the world of like, I don't know who the speaker is, this is a thing that will happen. And in some work led by Wes Gurney that we're going to talk about later, we found neurons that detected things like this, Texas and French. This Texas Python code. And in some sense this is like a particularly trivial world model. And so yeah, that's an interesting thing in my opinion. It was kind of a priori obvious that language models would learn this if they could if they could and needed to and it was more efficient.
SPEAKER: Tim Scarfe
Another point of order though, um, learning that something is French seems categorically different because when I, when I read Kenneth's original piece, he showed what looked like a topological representation of the world. So how different state spaces were related to each other in a kind of network structure. So I wonder if you can remember how we produced that diagram.
SPEAKER: Neel Nanda
Yeah. So I'm struggling to remember the details. I think it was something of the form. Look at how different cells are represented in the model and look at how close together the representations of different cells are. And oh, the model has kind of got internal representations that are close together. I don't think this is fundamentally different from the King queen man, woman thing of just it's like learn some structure and his representations. That's obviously kind of reasonable. Yeah, I yeah, I wouldn't read too much into that. Like models learn structural representations I think is old news at this point. Um.
SPEAKER: Tim Scarfe
But maybe another interesting angle is that one of the reasons why people like Gary Marcus, they say GPT is parasitic on the data. They say because they are empirical models, most of the meaning, most of the information is not in the data. We have to reason over explicit world models. So he thinks the reason a GPS is so good is because we've imputed this abstract world model. And similarly when we play chess we have an abstract world model. And he would argue that the information about that abstract world model doesn't exist in any data. So how do you go from the data to the model and the Othello game seem to show that you could go from the data to the model?
SPEAKER: Neel Nanda
Yeah, I know. I think that viewpoint is just like obviously wrong. Like you're trying you're trying to do a data prediction problem. A valid solution to that is to model the underlying world and use this to predict what comes next. There's clearly enough information in an information theoretic sense to do this. And the question is, is a model capable of doing that or not? And I don't know. I'm just like, you can't write poetry with statistical correlations. You need to be learning something. Maybe that's not a good example. I don't believe you can write like you can. I don't believe you can produce like good answers to like difficult codeforces problems. It's like do good software engineering as purely a bundle of statistical correlations. Um, maybe I have too much respect for software engineers. I don't know.

So where does it.
SPEAKER: Tim Scarfe
Come from then?
SPEAKER: Neel Nanda
Um.

SPEAKER: Tim Scarfe
That flash of inspiration or that high level? I guess the first question is do you is there a jump? Is it actually grounded in the data it's trained on or is there some high level reasoning? You know, where does that materialize from?
SPEAKER: Neel Nanda
So the way I think about it, there is just a space of possible algorithms that can be implemented in a transformer's weights and some of these look like a world model and some of these look like a bunch of statistical correlations. And models are trading off lots of different resources like how many dimensions does this consume? How much weight norm, how many parameters, how hard is this to get to and how weird and intricate and models will choose the thing that gets the best loss that is most efficient on these dimensions, assuming they can reach it within the loss landscape where use choose in a very anthropomorphic sense like Adam chooses good solutions and I don't know if you have a sufficiently hard task and forming a world model is like the right solution to it. Models can do it and I think people try to put all of these fancy philosophizing on it in a way that I just think is false.

Guilty as charged.
SPEAKER: Neel Nanda
And I think the Othello paper is like a really beautiful, elegant setup that proves this. All right. Can I move on to the plot twist?
SPEAKER: Tim Scarfe
Does it prove it, though?

Um, it's very.
SPEAKER: Tim Scarfe
It's a it's a very small, contrived. It's a big jump to assume that that works on a large language model.
SPEAKER: Neel Nanda
So this is kind of the argument I'm making. I think there's the empirical question of do language models do this and the theoretical question of could they do this? And I'm saying I think the theoretical question is nonsense. And I think the Othello paper very conclusively proves the theoretical question is nonsense, which is like, yeah, when given a bunch of data you can infer the underlying world model behind it. Well, in theory I.
SPEAKER: Tim Scarfe
Would push back on that a tiny bit because it's very similar to AlphaGo proved that in a closed game, which is systematic and representable um, you know with, with a finite obviously exponentially large but a finite number of board states you can you can build an agent which performs really, really well. That seems to me completely different to something like language or acting in the. Real world that might not be systematic in the same way. We can debate whether or not it's I think it's an infinite number of possible trajectories. Just like language, an infinite number of possible sentences.
SPEAKER: Neel Nanda
No, man. There's 50,000 to the power of a thousand possible infinite sequences. Sure is a finite number.

You mean in Othello or.
SPEAKER: Neel Nanda
No, no. In Gpt2 in GPT two.

Okay.
SPEAKER: Tim Scarfe
Uh.
SPEAKER: Neel Nanda
Bounded context length bounded vocab size more generally.

Bastard. You're not gonna.
SPEAKER: Neel Nanda
Write more than 1 quintillion characters? Probably.

Yeah.
SPEAKER: Neel Nanda
Being a parent, do you continue? Yeah.
SPEAKER: Tim Scarfe
Well, I guess it is still a big jump though, isn't it? From. Yes, Empirically it shows that in Othello it works. Maybe. Maybe we could debate whether or not it does or not because there's always this question coming back to what we were saying before, whether it's learning something which is universal or something which is still brittle. So the way that we've evaluated it might lead us to conclude that it's universal, whereas actually it's brittle in ways that we don't understand. So that's a very real possibility.
SPEAKER: Neel Nanda
Yeah. And like everything's brittle in ways you don't understand. It's like pretty rare that a model will do everything perfectly in a way that there are no adversarial examples. And this is like one of the more interesting things that's come out of the adversarial examples literature. To me it's just like, Oh wow, there's so much stuff here. There's so there's such a high dimensional input space. There's all kinds of weird things the model wasn't prepared for. And I don't know my interpretation of the Othello thing is the strong theoretical arguments are wrong. I separately believe the you know, there are world models that could be implemented in a language model weights, but they also disagree with the strong inference of the paper that this does happen in language models or that we conclude it does because what models are often really expensive like in the Othello model, it's consuming 128 dimensions of its 512 dimensional residual stream for this world model. And the problem is set up. So the world model is insanely useful because whether a move is legal is purely determined by the board state. So it's worth the model's while to do this. But this is rarely the case in language. For example, there was all this buzz about Bing chat playing chess and making legal ish moves. Yes, and I don't know man. If you want to model a chess board, you just look at the last piece that moved into a cell. That's the piece In that cell, you don't need an explicit representation. You can just use attention heads to do it. And there's all kinds of weird hacks and like models will generally use the best hack, but probably it is worth the model's while to have some kind of an internal representation. Like I'd bet that if you took a powerful code playing model and probed it to understand the state of the key variables, it would probably have some representation. But yes, moving on to the work I did building on the Othello paper. So one of the things that was really striking to me about the Othello work is simultaneously its results were strong enough that something here was clearly real, but they also used techniques that felt more powerful than when needed. Like rather they found that linear probes did not work. There weren't just directions in space corresponding to board states, but that nonlinear probes one hidden layer MLPs did. And the key thing to be careful

of when probing is is your probe doing the computation or does the model genuinely have this represented? And even with linear probes, there can be this can be misleading. Like if you're looking at how a model represents colored shapes and you find a red triangle direction, it could be that there's a red, green or blue direction and a triangle square or shape direction and you're taking the red plus triangle. Or it could be the case that each of the nine shapes has its own direction. You found the Red Triangle one, but nonlinear probing is particularly sketchy like in the extreme case if you train GPT three on the inputs to something deeply, three can do a lot of stuff. If you train GPT three on the activation side network, it can probably recover arbitrary functions of the input. Assuming the information on the input hasn't been lost which it shouldn't have because there's a residual stream. Um, and what I said is not quite true but not important. Um, and so I was and their intervention technique was both got like very impressive results but also involved doing a bunch of complex gradient descent against that probe. And this all just seemed more powerful than was necessary. And so I did the I challenged myself to do a weekend hackathon trying. To figure out what was going on and poked around at some internal circuitry and tried to answer some very narrow questions about the model and found this one neuron that seemed to be looking for like three cell diagonal lines where one was blank, the other was white, the next was black. But then sometimes it activated on blank, black, white. And it turns out that the general pattern was that it was blank. Current players. Sorry, blank opponent's color and current players color. And this is a useful motif in Othello because it makes the move legal. And when I saw this I made the bold hypothesis. Maybe the model actually represents things in terms of whether a cell has the current player's color or the current opponent's color, which in hindsight is a lot more natural because the model plays both black and white and it's kind of symmetric from the perspective of the current player. And I trained a linear probe on this and it just worked fabulously and got near perfect accuracy and I tried linear representations on it and I tried linear interventions and it just worked and I feel really excited about this project for a bunch of reasons. First lighted in a weekend. I'm still very proud of this. Secondly, I think that it has vindicated some of my general suspicion of nonlinear probing. Like if you really understand a thing, you should be able to get a linear probe to work and kind of more deeply as we discussed, there's this word two Vec style linear representation hypothesis about models that features correspond to directions The Othello works seemed like pretty strong evidence against. They had causal interventions showing that the board state was there but actually nonlinear but linear probes did not work. Seemed like they found some nonlinear representation and my and Chris Ola's hypothesis seeing this was that there was a linear representation hiding beneath Martin Wattenberg, one of the authors of the paper, had the hypothesis that it was like an actual nonlinear representation and this was evidence against the hypothesis. And this kind of formed a natural experiment where the hypothesis could have been falsified. But my work showed there was a real nonlinear, a real linear representation and thus that it had predictive power. And so many of our frameworks for Mcinturf are just these loose things based on a bunch of data but not fully rigorous or fully conclusively shown. And so natural experiments like this feel like some of the best data we have.

SPEAKER: Tim Scarfe

On this linear representation though I don't know if you've heard of the Spline Theory of Neural Networks by Randall Ballestrero and without going into too much detail, it's quite a discrete view of MLPs in particular that the Relu's essentially get activated in an input sensitive way to carve out these polyhedra in the ambient space and essentially an input will be mapped into one of these cells in the ambient space. And then there's a kind of

discreteness to it because if you just perturb the input and you move outside of one of these polyhedra, then the model will if it's a classical classifier, classify something different. But guess I want to understand with this representation theory. If features are directions, does that imply there's a kind of continuity because the network will learn to spread out those representations in the best possible way, but it won't necessarily be a way which is semantically useful. Like in Word two, Vec stop and go are very close to each other and they shouldn't be. And at what point does stop become go? So do you. Do you see there being boundaries in these directions?
SPEAKER: Neel Nanda
So I think this is again my point that I think of linear representations as being importantly different from geometric representations like stop should be close to go because in many contexts they are like a kind of changing of state term and it's used in similar contexts and has similar grammatical meaning. But then on this like single semantic thing they're like quite different And the natural way to represent this is have them be close together in Euclidean space but have some crucial like negation dimension with their difference and for context and like ultimately neural networks are not geometric objects. They are made of linear algebra. Every neurons input is just projects the residual stream onto some vector and this involves just selecting some set of directions and taking a linear combination of the feature corresponding to each of those. And this is just the. Natural way for a model to represent things, in my opinion.

Okay. Okay.
SPEAKER: Tim Scarfe
Well, I think this will in a second lead us on very nicely to superposition, which is that we don't actually think of there being one direction necessarily. Just just to close this little piece now, you said in your Lesswrong article that Othello GPT is likely over parameterized for good performance on this particular task while language models are under parameterized and of course we have the ground truth to this task which makes it very, very easy.
SPEAKER: Neel Nanda
So much easier to interpret 100%.
SPEAKER: Tim Scarfe
But but you did you did conclude saying that this is further evidence that neural networks are genuinely understandable and interpretable and probing on the face of it seems like a very exciting approach to understand what the models really represent. Caveat emptor, conceptual issues. So let's move on to to this superposition also known as Poly Semanticity, which is an absolutely beautiful well, you're shaking your head a little bit, so maybe maybe you start with that.
SPEAKER: Neel Nanda
Um, yeah. So there's. All right, so what's the narrative here? So fundamentally we are trying to engage with models as these high dimensional objects in kind of this conceptual way. So we need to be able to decompose them because of the curse of dimensionality. And we think models correspond to features and the features correspond to directions. And the hope in the early field was that features would correspond to neurons. And even if you believe features correspond to orthogonal directions, the same that they correspond to neurons is like a pretty strong one because a priori there's no reason to align with the neuron basis. The reason this isn't a crazy belief is that models are incentivized to represent features in ways that can vary independently from each other. And because Relu's and Jellies act element wise, if there's a feature per neuron, they can vary independently. Well, if there's multiple

features in the same neuron, I don't know if there's a Relu. The second feature could change. So the relu goes from on to off in a way that changes how the other feature is expressed in the downstream network. And this is like a beautiful theoretical argument. Sadly it's bullshit because of this phenomena of poly Semanticity poly Semanticity is a behavioral observation of networks, but when we look at neurons and look at things that activate them, they're often activated by seemingly unrelated things like the ERs in the word strangers and capital letters are proper nouns and news articles about football adds a particularly fun neuron I found one time in a language model and um poly semanticity is a purely behavioral thing. We're just saying this neuron activates for a bunch of seemingly unrelated stuff. Um, and it's possible that actually we're missing some galaxy brained abstraction where all of this is related. But my guess is that this is just the model is not aligning features with neurons. And one explanation of this is you've just got this thing called a distributed representation where a feature is made of a linear combination of different neurons, but it is kind of rotated from the neuron basis. And this argument that neurons can vary independently is a reason to think you wouldn't see this um, where this hypothesis is just that there's still n things when there's n neurons, but they're rotated. Um, but then there's this stronger hypothesis that tries to explain this called the superposition hypothesis. And here the idea is so if a model wants to be able to recover a feature perfectly, it must be orthogonal from all other features. But if it wants to mostly recover it, it suffices to have almost orthogonal vectors and you can fit in many, many more almost orthogonal vectors into a space than orthogonal vectors as theorem saying that there are exponentially many in the number of dimensions.
SPEAKER: Tim Scarfe
How if you have 100 dimensional vectors, how many orthogonal directions are there? What's the relationship? 100?

Yep.
SPEAKER: Neel Nanda
Um. Yeah. This is just the statement that like you pick one, you pick a vector. Um, sorry. There's 100 vectors that are all orthogonal of each other. Um, basic proof. You pick a vector, everything's orthogonal so that that's a 99 dimensional space. You pick another vector, take everything orthogonal to that, that's a 98 dimensional space and keep going until you get to nothing. Um, like if you picture a 2D space, you pick any direction. The only things orthogonal to that are a line. And so there's exactly two orthogonal things you can fit in and there's. Like you can rotate this and you can get many different sets of orthogonal things.
SPEAKER: Tim Scarfe
Okay. I'm trying to articulate why this doesn't make sense to me. So maybe we should start with the curse of dimensionality, which is that the volume of the space increases exponentially with the number of dimensions. So we'll start with that. And the reason I'm thinking maybe maybe I'm wrong, but if you've got 100 dimensional vector, um, every combination of flipping one of the dimensions would be would produce a vector which is orthogonal to all of the other ones.

Would it not?
SPEAKER: Neel Nanda
Uh, no. So let's imagine you've got a vector of all ones if you pick the first element and negate it. Yeah. So it's like minus one then 99 ones. These are not orthogonal. The dot product is 98.

Okay.
SPEAKER: Tim Scarfe
Okay. Well, that makes sense. So. So there's there's a linear number of orthogonal directions and in which case we actually need to have these approximately orthogonal directions because that actually does buy us an exponential number.
SPEAKER: Neel Nanda
Yeah. And so the superposition hypothesis is that the model represents more features than it has neurons. Yes. Or that it has dimensions and it somehow compresses them in as things that are almost orthogonal. When it reads them out with a projection, it gets some interference. But the and it needs to balance the value of representing more features against the costs of interference and anthropic has this fantastic paper called Toy Models of Superposition, which sadly was written after I left, so I can't claim any credit. And what they they basically build a toy model that exhibits superposition the exact structure as you have n independent features, each of which is zero most of the time it's not very prevalent and there's a linear map from that to a small dimensional space, a linear map back up and a non-linearity on the output. No non-linearity on the input on the bottleneck in the middle and you you train it to be an auto encoder. Can it recover the features in the input And because there's many more features than there are in the bottleneck this tests whether the model can actually do this and they find that it sometimes does sometimes doesn't and then do a lot of really in-depth investigation of how this varies. And yeah, returning to like is superposition the same thing as poly semanticity? Um, I would say no policeman taste is a behavioral thing. Distributed representations are also a behavioral thing that it's like not aligned with the basis and superposition is a mechanistic hypothesis for why both of these will happen. Because if you have more features than neurons, obviously you're going to have multiple features per neuron and probably you're going to have features that are not aligned with neurons.
SPEAKER: Tim Scarfe
Okay. Okay. Very interesting. So why do you think that superposition is one of the biggest problems in McInturff?
SPEAKER: Neel Nanda
Yeah. So it's this fundamental thing that we need to be able to decompose a model into individual units and ideally these would be neurons, but they are not neurons. So we need to figure out what we're doing and superposition. So in a world where we just had like an a meaningful directions, but they weren't aligned with the standard basis, that would be kind of doable. Um, and indeed models often have like linear bottlenecks like the residual stream or the keys queries and values of an attention heads that don't have element wise non-linearities and so have no intrinsically meaningful basis. The jargon here is privileged basis and but superposition means that you can't even say this feature should be orthogonal to everything else. There's going to be a bunch of interference. Um, there's not even a kind of mathematically, mathematically, there's not even like a unique set of more than n directions to describe some set of vectors in n-dimensional space. Um, and I think that understanding how to extract features from superposition given that superposition seems like a core part of how models do things. Though we really do not have as much data here as I would like us to. Um, understanding how to extract the right meaningful units seems really important.
SPEAKER: Tim Scarfe

Okay. And I think we should clarify the difference between computational and representational superposition.
SPEAKER: Neel Nanda
Yeah. So there's kind of two. So transformers are interesting because they often have high dimensional activations that get linearly mapped to low dimensional things. So like in say, GPT two. In say, GPT two small, the residual stream has 768 dimensions, while each MLP layer has 3000 neurons. And even if we think each neuron just produces a single feature, they need to get compressed down to the 768 dimensional residual stream and we or there's like 50,000 input tokens that get compressed to 768 dimensions. And this is called representational superposition. The model is representing the model's already computed the features, but it's compressing them to some bottleneck space. And this is the main thing studied in the toy models of superposition paper and what we found. Sorry. Um, there's a separate thing of computational superposition, which is when the model is doing, it's computing new features. This needs non-linearities like attention head softmaxes or MLP. Jealous and the Non-linearities can compute new features as directions from the old ones. Like um if this for example, if the top of an image is a car window and the bottom is a car wheel, then it's a car. Um, or if the current token is Johnson and the previous token was Boris, this is Boris Johnson and this is all. How to phrase this? Um. Yeah, this is computational superposition. If the model wants to compute more features than it has neurons and this is like much harder to reason about because linear algebra is nice and fairly well understood. Non-linearities Spoilers in the name are not linear and thus way more of a pain. And I think that we generally have a much less good handle on computational superposition, but also that this is like way more of where the interestingness lies by my lights and this is very briefly studied in the toy models of superposition paper but would love to see more work looking at this in practice and also looking at this in toy models.
SPEAKER: Tim Scarfe
So zooming out a tiny bit, there's this paper from anthropic and the overall question to me is does it actually exist? Now presumably you're satisfied with the evidence that it does exist and then there's the question of how do neural networks actually do it? And then there's the question of how does the neural network think anthropomorphic language? I apologize about the trade off of more super more superposition, more features, but more interference versus less interference and more superposition.
SPEAKER: Neel Nanda
Yeah. So diving into the final question about interference, um, the a useful conceptual distinction is that there's two different kinds of interference. So if you've got two features that share a dimension or share a neuron, Um oh yeah. A final note on representational superposition is I don't think it should even be referred to in terms of neurons because the individual basis elements don't have intrinsic meaning, modulo weird quirks like atom. Um, and it annoys me when people refer to the residual stream or key vectors as having neurons. There's no element wise linearity. It's not privileged anyway. Um, yeah. Two types of interference when A and B share a dimension you can. Yeah. Let's say this dimension has both dice and poetry. You first off, need to tell where if dice is there. But poetry is not. You need to tell that dice is there and that poetry is not there. And if both which I call alternating interference and then there's simultaneous interference where dice and poetry are both there and you need to tell that both are there, but not that they're both there with like double strength. And as a general rule, models are good at dealing with things of the form. Notice when something is extreme along this dimension but not notice when it is extreme along a dimension versus when it's not extreme and alternating alternating interference looks like

that. Like if dice is straight up, poetry is at 45 degrees. Both have like weak inter, both have less interference when the other one is active than when the main one is active along their direction.
SPEAKER: Tim Scarfe
Okay, so you're saying interference from A and not B is far easier than A and B?
SPEAKER: Neel Nanda
Yes, exactly. And like a very a very rough heuristic as models will just not do simultaneous interference but will do alternating interference. And they observed this in the toy models paper because they varied how often a feature was non-zero. What I think of as the prevalence of the feature though they called it sparsity. And what they found is that when the feature was less prevalent, it was much more likely to be in superposition. And the way to think about this is if you have two independent features that both exist with probability. P the rate of simultaneous interference is P squared. The rate of alternating is P and so and the worth of having the feature is also proportional to P because it occurs P of the time. So the rarer it is, the less of a big deal simultaneous interferences and eventually the model uses superposition. There's also there was also an interesting bit looking at correlated features. So correlated features, even if they're not very prevalent, they have pretty high simultaneous interference and models tend to put correlated features in to be orthogonal. But anti-correlated features, it's very happy for them to share a direction. One way you could think about this is if you've got say, 25 features about romance novels and 25 features about Python code, you can have 25 directions that each contain a pair of features and then a single disambiguating neuron that is on for Python code off of romance novels that you use to disambiguate the two. And yeah, maybe this would be a good time to talk about the finding neurons in a haystack paper. Or unless you've got more stuff on this.
SPEAKER: Tim Scarfe
We'll get to that in just two shakes of a lamb's tail. But just before when I was reading through the paper, I was I had the mindset of sparsity. And you told me, Tim, don't don't say sparsity. It's prevalence.
SPEAKER: S4
It means so many things.
SPEAKER: Tim Scarfe
It means so it's very overloaded.
SPEAKER: S4
Such an overloaded word.
SPEAKER: Tim Scarfe
So, you know, so just quickly touch on the relationship between or what what is prevalence, the relationship between prevalence and superposition And just before Well, actually, I've got a couple more questions, but would you also just mind playing devil's advocate and criticizing the anthropic paper if you can?
SPEAKER: Neel Nanda
Sure. So I should be very clear. This is one of my top three all time favorite interpretability papers. It's a fantastic paper that said.

A bad word said about it.
SPEAKER: Neel Nanda
Um oh, I have so much I have bad words to say about every paper, especially the ones that I like because I've engaged with them in the most detail. So things which I think were misleading about this paper. The first is I think the representational versus computational

superposition distinction is very important. I think computational is a fair bit more interesting. And while I think the authors knew the difference, I think a casual reader often came away not realizing the difference in particular that most of their results were about the residual stream, not about actual neurons and layers. Um, the second is a question of activation range. So they study features that vary uniformly between 0 and 1. And in practice I think most features are binary. This is a car wheel or this is not a car wheel. This is this is Boris Johnson and this is not Boris Johnson. And interference is much worse when they can vary continuously because if A and B if A is up B is at 45 degrees, you can't distinguish B at strength one from A at strength 0.7 ish. And this is just kind of messy, but at that binary it's just much easier. And I think this is a source of confusion. Um, yeah. I also think the two kinds of interference point was a bit understated and yeah, but like more broadly it was just a phenomenal paper. Oh my other biggest beef that they just didn't look in real models and like this wasn't the point of the paper, but like, Oh, we're doing so much theory crafting and forming conceptual frameworks and we haven't really checked very hard whether this is why models actually have semanticity.

SPEAKER: Tim Scarfe

Um, whereas Gurney, he's working out of MIT and you've done a lot of work with him. So, uh, you and Wes. Wes was the first author wrote a paper called Finding Neurons in a Haystack Case Studies with sparse probing where you empirically studied superposition in language models and actually found that you get lots of superpositions in early layers for features like the security and Social Security and fewer in middle layers for complex features like this text is French. So and also you can bring in the importance of range activations as well. But can you frame up that paper?

SPEAKER: Neel Nanda

Yeah. So first off, this paper was led by Wes Gurney. One of my mentees did a fantastic job. He deserves like 9% of the credit. Great job, Wes. Uh, I believe he listened to this podcast, so. Hi. Um, and yeah, so the kind of high level pitch behind the paper was, well, we think superposition is happening, but like nobody's really checked very hard and there's like some results in the literature I've since come across in non transformer models that demonstrate some amount of distributed representations. But what would it look like to check and what would it look like to do this in like a reasonably scalable and quantitative way And the kind of sparse probing in the title is this technique Wes introduces for um, if we think of feature as represented in MLP layer, we can train a linear classifier to extract it, a linear probe from that layer. But if we constrain the probe to use at most k neurons vary K and look at probe performance. This lets us distinguish between features that are represented with like a single neuron and features that are densely spread across all neurons with a lot of mythological nuances about balanced data sets and avoiding overfitting and fun stuff like that. And most of the interesting bits of the paper, in my opinion, are the various case studies we do where so probing fundamentally is like a kind of sketchy methodology. Because probing is correlational. Probing doesn't tell you whether a model uses something. And it's so easy to trick yourself about whether you have the right representations. So we use it as a starting point and then dig more deeply into a few more interesting things. One particularly cute case study is we looked into factual knowledge neurons found something that seemed to represent this athlete, plays hockey but then actually turned out to be a Canada neuron which continues to bring me joy that activates with things like maple sirup and Canada. Wonderful. Got Gotta Love models learning national stereotypes, right? Oh yes. Um, anyway, so, um. Yeah. So there were two particularly exciting case studies. The first was looking in early layers at compound word detectors. So if you look at, say, the brain and its

visual field, we have all these sensory neurons. We get raw input of light from the environment and it gets converted into stuff our brain can actually manipulate. Image models have Gabor filters that convert the pixels into something a bit more useful. What's the equivalent of language models? And it seems to be these things that we call tokenization neurons and circuitry where often words are split into multiple tokens or you get common compound word phrases like Social Security or Theresa may or Barack Obama and whatever. And it's often useful for a model to realize this is the second thing in a multi token phrase, especially if it's like you need both things. Know what's going on. Like Michael Jordan, the Michael's lots of Jordans. It's really important to tell the both of them are there and this is a clearly nonlinear thing because it's like a Boolean and and so we did a lot of probing for different compound words and we found that they were definitely not represented well by single neurons. We could find some neurons that were okay at detecting them, but there was a lot of interference and a lot of like false positives from other stuff. And when we dug into a bunch of these neurons, we found that they were incredibly poly semantic. They activated for many different compound words and we showed that it was using superposition by observing that if you took, say, five Social Security detecting neurons and add together their activations, they go from okay detectors to a really good detector together because even though each is representing like hundreds of compound words, they're representing different compound words which lets you encode these And this what we've shown here is that it's like distributed, that it's a linear combination of neurons. We still haven't shown it perfectly to my dissatisfaction. I think you really need to do things like oblate, these linear combinations and see if this systematically damages the model's ability to think about Social Security, etcetera. But I'm pretty convinced at this point and there's like a few properties of compound words that both make it easy to represent in superposition and make me pretty okay making the jump that there's actual superposition. The first stage is that there's tons of compound words. Each one is pretty rare, but each one is like nontrivially useful and clearly there are more compound words than there are the like thousands of neurons in the layer of this model. The model cares about representing and can represent that we do not actually check because I could not convince Wes to accumulate a list of 2000 compound words and probe for all of them. Um, but I believe in my heart this is true.
SPEAKER: Tim Scarfe
Could I have a point of order though? So go for it. Because I've been reading quite a lot of stuff from, um, linguists like Steven Piantadosi and a lot of linguists are some of them hate language models and some of them are well on board with it. And you know, like Raphael Milliere, for example, is a great example.
SPEAKER: Neel Nanda
Um, I hate language models too, don't worry.

Well, but.
SPEAKER: Tim Scarfe
But the question is because you're talking about compound words and stuff like that and you're still using the language of syntax and these language models, there's this distribution hypothesis, you know, you know a, you know the meaning of a word by the company it keeps. But linguists and cognitive scientists kind of ditch that. I think I don't think they ever believed in the distributional hypothesis. They think about grounding, They think about grounding two things in the world and and and also inferential references as well, which is you can think of that as grounding to a model of the mind. And this brings us back to the Othello paper, which is that they're not just learning simple kind of compound relationships.

Ships between the world. Between the words they're learning a world model and they're doing something much more potentially than just predicting the next word. And Piantedosi argued that most of the representational capacity in language models are learning these semantics. They're learning relationships between things in the world model and the particular occurrence of of the token. And this superposition idea is very interesting because it actually imbues the representational capacity in a language model to learn those mappings. Um.
SPEAKER: Neel Nanda
Uh, okay. So a couple of comments on that. The first is a generally useful way of thinking about models to me is as a the early layers devoted to sensory neurons, converting the raw input into more useful concepts and representations, the actual processing throughout, like all of the middle layers that actually does all the reasoning and then motor neurons at the end that convert the reasoning to actual output tokens for like the format that the optimizer wants. And it feels like you're mostly talking about the like reasoning internally and I'm the specific case that I'm referring to is on the sensory neurons. Well, like I'm not saying it just detects compound words, but obviously that's the first thing it does.

I don't know.
SPEAKER: Tim Scarfe
It's so interesting. I mean, I don't mean to push back, but in neuroscience, the field was held back for decades by this idea of this kind of left to right processing, this hierarchical processing where you have these very, very simple concepts that become increasingly abstract with more processing. And then I think the field has moved away from that. It's far more messy and chaotic than now with the neural network. It actually is hierarchical because the network is basically a Dag. So I suppose it is safe to make this assumption. But could I just kind of question you on that? Is it safe to make that assumption? Is there increasing complexity in representation as you go from left to right?
SPEAKER: Neel Nanda
Uh, let's see. Uh, so yeah, definitely. Yeah. So clarification one the network has this input sequence which I think is going from left to right and then there's a bunch of layers which I think of as going from like the bottom to the top. Yes. And you're referring to the bottom to top axis, right?
SPEAKER: Tim Scarfe
Yeah. I'm sorry. I was using an MLP mindset when I asked that question. So as you say, in a transformer, it's an Autoregressive model and you have, you know, stacked attention layers with little MLPs on the end. So I guess the way I was actually meaning the question is so, so, so complexity increases monotonically as you go up the stack of attention layers. Is that a fair assumption?
SPEAKER: Neel Nanda
Um, yeah. Uh, again, no one's really shown this properly, but I'm like, Surely this is true And there's been some work doing things like looking at neurons, looking at the text that activates them, looking for patterns and trying to understand what, what these represent. And it's generally looks like early ones are more about tokenization and syntax. Later ones are doing stuff that's interesting. Final ones are doing this like motor neuron behavior. But like I also want to be very clear that networks are coerced, networks do not fit into nice abstractions. I'm not saying the early layers are literally only doing tokenization. Yeah, but I believe we have shown it's part of what they're doing and I speculate it is a large part of what they're doing. I'd be very surprised if it's all of what they're.

SPEAKER: Tim Scarfe
Doing because I heard you on another podcast and it you were just talking about the I mean, I think the curse is the right way to describe it, which is that even when you make, um, modifications, when you manipulate what's happening, the behavior will change in a very reflexive way. So you kind of you delete one thing and then another neuron will take on the responsibility of the thing you just deleted. And so, so you're it's a little bit like manipulating financial markets. You've got almost like this weird collective diffuse intelligence where you make one modification and the whole thing changes in a very complex way. And similarly, I guess that's why was intuitively questioning the assumption that you have a residual stream. So surely even at the very top of that attention stack, there must be primitive and complex operations going on in some weird mix.
SPEAKER: Neel Nanda
Um, it seems probably true generally, yeah. There's going to be some stuff you can just do with literally the embeddings. Um, some stuff that you need to wait a bit more before you can do anything useful with. Just like I know if you got a sentence about Michael Jordan, I don't think he can usefully use Michael Jordan in isolation. So you need to detokenize to Michael Jordan. But also I don't know if you've got Barack Obama a bomber and Barrack both on their own pretty clearly imply it's going to be about a. Bomber and probably the model can start doing some processing in the early. Like layer zero. Does it want to? Somewhat unclear. It's going to depend a lot on the model's constraints and other circuitry and how much it's worth spending the parameters then versus later. There's also some various things where don't know, model memory kind of decays over time because the residual streams norm gets bigger. So early layer outputs become a smaller fraction of the overall thing and layer norm sets the norm to be units. So things kind of decay. And so if you compute a feature in the early in like layer zero, it can be harder to notice by like layer three than if it was computed in layer two. But these are all just kind of like mild nudges and ultimately neural net could do what neural networks want.

Man I know, I know.
SPEAKER: Tim Scarfe
I just want to close the loop on something said a little while ago about, you know, potentially large models use most of their representational capacity for, um, you know, learning these semantic relationships. And empirically we found that, you know, there's some question recently actually about do we actually need to have really, really large models and for pure knowledge representation the argument seems to be yes, but we can disentangle knowing from reasoning and there's also this mimicry thing. So it's quite interesting that all of the you know, like Facebook released their model and very, very quickly people fine tuned it using the Laura you know, the low rank approximation fine tuning method and on all of the benchmarks the model I mean even open assistant is another great example. Yannick was sitting in your seat just a few weeks ago and he was saying that on on many of the benchmarks the is working really well. But it's kind of not it's kind of mimicry like the big large models that you know meta and Google and DeepMind and all these people they spend millions training these models and they have they have base knowledge about the world, which is not going to be, you know, replicated by fine tuning, you know, like an open source model anytime soon.
SPEAKER: Neel Nanda
The knowledge is based.

It's okay.
SPEAKER: Neel Nanda
The knowledge is based.

Yes, yes.
SPEAKER: Tim Scarfe
Yes, yes, exactly. Well, um, okay, so so that's that's that's very interesting. Let's just quickly talk about the OpenAI microscope because this is the OpenAI microscope is this beautiful app that OpenAI released in 2020. And you can go on there and you can you can click on any of the neurons in popular vision architectures at the time. So I think most of them are sort of like image net, you know, things like alexnet and God knows what else. And they they solve this optimization problem where they generate an image using stochastic gradient descent that maximally activates a particular neuron or I think even a layer using something similar to deep dream. And you can click on these neurons and sometimes they are what we will call poly sort of mono semantic, which means it's just Canada a lot of the time there's a couple of concepts in there, but it's weirdly intelligible, you know, you might see, you know, like a playing card or an ace and a couple of like tangentially related concepts. And it always struck me as strange because I imagine there's a long tail of semantic relationships. And I found it bizarre that there'd only be 1 or 2 in this visualization. And I had this intuition that the optimization algorithm is in some sense mode seeking rather than distribution matching, which is to say that it finds the two most or two or 3 or 4 most kind of salient semantic mappings and they dominate what is visualized and you almost snipping off the long tail of of the other semantic mappings.
SPEAKER: Neel Nanda
Yeah. So I think there's two things to disentangle here. The first is what is actually represented by the neuron in terms of ground truth and the second is what our techniques show us. So the two techniques used in the OpenAI microscope are looking at the images, the most activated neuron and then this feature visualization technique where they produce a synthetic image that maximally activates it. And to me this is these are like both of these can be misleading because if the model activates the dice and poetry but activates the dice with Trent five and poetry Trent four, then the optimal image activated will be dice and the optimal the data set examples will also be dice, but really it'll be about poetry and you want to get a lot more rigorous if you want to show true mono Semanticity Um, one cute thing is spectrum plots. You take lots of example data set examples across the full distribution. You have a histogram with like the different groups for the different meanings and then neuron. On the x axis. We had this really cute plot in Wes's paper called the French Neuron where all of the French were. All of the French text is on the right. All the non French text is on the left and then you're on it. Just very clearly distinguishing the two in a way that's much more convincing to me than things like Max Act examples. Um, and I actually have a hobby project called Neuro scope at neuro scope where you can see the max activating text examples for every neuron and a bunch of language models though OpenAI recently output this paper with one that is just better but only for GPT two XL. Um. Anyway, not that I'm bitter or anything so. Um. And yeah. So yeah, there's the things can lie to you and be illusory. Um, there's this interesting paper called The Interpretability Illusion for Bert which investigated this specific phenomena and in particular that if you take the dataset examples over some narrow distribution like Wikipedia or books, you can get pretty misleading things though they only looked at residual stream basis elements rather than actual neurons I believe, which makes it a bit less compelling.

Um, point of order as well.
SPEAKER: Tim Scarfe
We've been saying residual stream quite a lot and Microsoft introduced Resnet in 2015, which basically means that between all of the layers the information is being passed up unadulterated so the subsequent layer can choose to either essentially shortcut or ignore the previous layer or use some combination. And at the time they kind of said it was about the neural network being able to learn its own capacity in some sense. But could you just give us like the way you think about these residual streams?

Yeah.
SPEAKER: Neel Nanda
So I think the standard view of neural networks is there are just layers and layer five's output is layer six is input, etcetera. Um, then people added resnets where layer six is input is layer five outputs plus layer five's inputs with the skip connection. But I think people normally thought of them as like a it's like a cute trick that makes the model better but doesn't massively change my conceptual picture and the framing that I believe was introduced in the mathematical framework. This anthropic paper led by Chris Ola Nelson Al-Haj and Katherine Olson that I was involved with is actually let's call the thing in the skip connection, the residual stream and think of it as the central object and draw our model. So the residual stream is this big vertical thing and each layer is like a small diversion to the side rather than the other way around. And in practice most circuits involve things skipping many layers and each layer should is better thought of as like an incremental update. And there's a bunch of earlier transformer interpretability papers that I think miss this conceptual point like the interpretability illusion for Bert what I mentioned earlier and study residual stream basis elements as like layer outputs or something.

Yeah.
SPEAKER: Tim Scarfe
I mean in a sense, you know, we were talking about being able to reuse things that you've learned before and not having to learn them again. And guess I think of it as a kind of translational equivariance in the in the layered regime, which is that you have a computation which is learned early on and now it can just be composed into subsequent layers. It's it's like you've got a menu of computational functions that you can call on at any layer.
SPEAKER: Neel Nanda
Yeah, pretty much. I think of it as like the shared memory and shared bandwidth of the model.

Yeah. Yeah.
SPEAKER: Tim Scarfe
Almost like a memory bus.
SPEAKER: Neel Nanda
Yeah. And sometimes models will dedicate neurons like cleaning up the memory and deleting things that are no longer needed.

Yeah. Yeah.
SPEAKER: Tim Scarfe
And is there any interference in that memory bus?

SPEAKER: Neel Nanda
So much. Go on. This is the thing of superposition, right? Like the residual stream is doing everything. There's 50,000 input tokens for start and then for x as many neurons as residual stream dimensions in every MLP layer and attention heads moving everything around. And it's just a klusterfuk.

What if.
SPEAKER: Tim Scarfe
You scale up the bandwidth of the.

Bus?
SPEAKER: Neel Nanda
Um, this is basically making the model bigger, right? Which we know makes models better.
SPEAKER: Tim Scarfe
But I don't know, just thinking out loud. But what if you maintained the original dimensionality of the model but you deliberately upscaled the bus?
SPEAKER: Neel Nanda
Um, so like, you make the thing inside each layer smaller but make the residual stream.

Bigger or.
SPEAKER: Tim Scarfe
Just make everything the same as it is, but you just kind of like have a linear transformation on the bus and double the size of the bus.

Um.
SPEAKER: Neel Nanda
So I don't think that would work without increasing the number of parameters because like if you because like the thing that matters is the smallest bottleneck, the output width of an MLP layer are like 4000 by 1000. And in order to make the 1000 bigger, you need more parameters. And there's like all kinds of studies about the optimal hyper parameters and the optimal ratios. My general heuristic is number of parameters are the main thing that matters. I don't know. I didn't spend that much time thinking about how to make models better, to be honest. I just want to understand them. God damn it.
SPEAKER: Tim Scarfe
Yeah, because it's one of those things that it might remove bottlenecks because. Because essentially you're allowing the model to reuse things that it's learned previously. So now every single layer can specialize more than it did before and that might kind of like weirdly remove bottlenecks.
SPEAKER: Neel Nanda
Yeah. Yeah, the way I generally think about it is models are ensembles of shallow paths, which is this paper from like five years ago about resnets like deep to small as 12 layers. Each layer includes an attention block and an attention bit an MLP bit. It is not the case that most computation is 24 levels of composition deep. It is the case that most of them involve like I don't know, four and they're just intelligently choosing which four and remixing them in interesting ways And sometimes different things will want to like get to different points. And so it's useful to have many layers rather than a few. But also I don't know if you halve if you halve the residual stream width and give the model for as many layers, often performance is like about the same or like not that different because the number of parameters is

unchanged and this is just kind of a wild result about models that I think only really makes sense within this framework of it's like an ensemble of shallow paths and it's a trade off between having more computation and having better memory bandwidth.

Yeah. Yeah.
SPEAKER: Tim Scarfe
Very interesting. Okay. I mean, just to close, um, superposition, it might not be a new idea. So Yannick did a, um, a paper video about this paper called Super Masks in Superposition by Mitchell Wartzman back in 2020 and he was talking about super masks representing sparse subnetworks in respect of catastrophic forgetting and continual learning. But that was slightly different because that was an explicit model to perform masking create subnetworks and to to model, you know, like basically a sparsity aware algorithm. But he was still using a lot of the same language like interference and so on and thinking about superpositions of subnetworks. And I guess the difference is, is like just as we were talking about with these inductive priors like Transformers and CNNs and the models already do this stuff without us having to explicitly code it, which I think is the interesting discovery.
SPEAKER: Neel Nanda
Yeah. Yeah. One update I've made from Wes's work is that D tokenization is probably like a pretty big fraction of what the earlier layers do and it's just really easy to represent compound words in superposition because it's a very binary. It's either there or not there. So alternating indifference is easy to deal with. They're mutually exclusive, so there's no simultaneous interference. Like you cannot have Boris Johnson and Theresa may co-occur. Um, and you there's just like so many of them. Um, one fact about language models that people who haven't played around them may not appreciate is their inputs are these things called tokens? And tokenizers are fucked because they're trained in this bizarre Byzantine way. That means that often words the rarer words will get broken up into many tokens. Yes, multi-word phrases are always different tokens. Anything that's weird like a URL gets completely coerced and models don't want to have this happen, so they devote a bunch of parameters to build a like pseudo vocabulary of what's going on. And just returning to your point earlier about like is it just these syntax level things? Is there some like actual more semantic stuff going on? Um, we did also have case studies looking at contextual neurons. Things like this code is in Python, this language is in French and these were seemingly mono semantic like it seemed like they were specific neurons here. And we found things like if you are bleat the French neuron loss on French texts gets much worse while other ones are fine. And also some interesting results that the model was say using this disambiguate. Things like tokens like D are common in German and also common in Dutch and the neurons for those languages were being used to disambiguate for that token, whether it was like a German or a Dutch D because they've got very different meaning in the two languages.

Yeah.
SPEAKER: Tim Scarfe
I wondered if you can give me some intuition like because as you say in Wes's paper, you know, he did actually find that there are some mono semantic neurons like French as you just said. And in this case, the model decided that interference in some sense wasn't worth the burden. But what does burden mean here? And French is a very vague concept as well.
SPEAKER: Neel Nanda

Yes. So All right. A couple of observations. First is I do not think we have properly shown they are mono semantic neurons. Um, we were looking these models were trained on the pile and we were specifically. Are You're looking at them on Europol, which is like a data set of European Parliament transcripts which are labeled by language. And we found a neuron that seemed to strongly disambiguate French from non French, but it was on this domain of parliamentary stuff. And because models really want to avoid simultaneous interference, if they did have superposition, they probably want to do it with something that isn't likely to co-occur in this context. I don't know. This is a list variable in Python which we didn't check very hard for. And in particular this is messy to check for because in order to do that you need to answer these questions like what is French? Like there's a bunch of English checks that will activate for but it will activate on words like Sacrebleu and Trebia. And I think I count this as French, but like I don't have a rigorous definition of French and I think an open problem I'd love to see someone pursue is just can you prove one of these neurons is actually a French detecting neuron or not? And what would it even look like to do that? And yeah, regarding interference in the burden. So the way I think about it, if two features are not orthogonal then um. Oh no, sorry, this is more interesting. In the case of neurons, if there's multiple things that could all activate a neuron, then it's harder for the downstream bit of the model to know how to use the fact that that neuron activated because there are multiple things even if they don't co-occur because they're mutually exclusive and this is just a cost and there's a trade off between having more features and not having this cost and features like this is in French are really load bearing. They're just really important for a lot of circuitry here. And so theoretically the model might want to dedicate an entire neuron to this, but if you dedicate an entire neuron, you lose the ability to do as much superposition. My intuition is the number of features that can be represented in superposition is actually like grows more than linearly with the number of dimensions. So this might be like significantly worse than just having one fewer feature.
SPEAKER: Tim Scarfe
So we are now in the next chapter of this beautiful podcast and we're going to talk about Transformers. So how exactly do Transformers represent algorithms and circuits? And also you've written this beautiful mathematical framework about Transformers, which of course is working very closely with Catherine Olson and Chris.

Oehler and.
SPEAKER: Neel Nanda
Nelson Al-Haj.
SPEAKER: Tim Scarfe
And and Nelson My.

Apologies.
SPEAKER: Neel Nanda
Um, yeah. So in terms of understanding, yeah. So if you want to do mechanistic interpretability on a model, you need to really deeply understand the structure of the model. What are the layers, what are the parameters, how do they fit together? What are the kinds of things that make sense there? And let's see. So. Yeah, there's like a couple of key things I'd want to emphasize from that paper, though. I don't know. It's also one of my like all time top three interpretability papers. People should just go read it and after reading it check out my three hour video walkthrough about it, which apparently is most useful if you've already read the paper because it's that deep anyway. Yeah. So a couple of things I'd want to call out

from that, especially for people who are kind of familiar with other networks but not Transformers The first we've already discussed the residual stream as the central object and the second is how to think about attention because attention is the main thing, which is weird about models. They have these layers which actually represent like two thirds of the parameters in a transformer, which is often an underrated fact. But attention is the interesting stuff. So Transformers have a separate residual stream for each input token and this contains like all memory the model wants to store at that position, but MLP layers can only process information in place. You need attention to move things between positions and classically people might have used stuff like a 1D convolution. You average over ten things in a sliding window. This is baking in the inductive bias that nearby information is more likely to be useful. But this is kind of a pretty limited bias to bake in. And the story of deep learning is that over time people have realized, Wait, we should not be trying to force the model to do specific things. We understand we should not be telling the model how to do its job. If it has enough parameters and is competent enough, it can figure it out on its own. And so the idea here is rather than giving it a convolution, you give it this attention mechanism where each token gets a query saying what it's looking for. Each previous token gets a key saying what it has to offer and the model looks from each destination token to the source tokens earlier on with the keys that are most relevant to the current query and models and the way to think about an attention head. So attention layers break up into these distinct bits called heads which act independently of the others and add to their outputs together and just directly add to the residual stream. This is sometimes phrased as concatenate the outputs and then multiply by a map. But this is mathematically equivalent. Um the each head acts independently and in parallel and further you can think of each head as separately breaking down into a which information to move a bit determined by the attention which is determined by the query and key calculating matrices and the what information do you move? Once I know where I'm looking, which are determined by the value and output matrices, we often think about these in terms of the QQ matrix WQ times W transpose and the matrix w o times w WV because there's no non-linearity in between. And these two matrices determine like what the head does. And the reason I say these are kind of independent is that once the model has decided which source tokens to look at, the information that gets output by the head is independent of the destination token and like the query only matters for choosing where to move information from. And this can result in interesting bugs. Like there's this motif of a skip trigram. The model realizes that. Hmm. If the current thing is three and two has appeared in the past then four is more likely to come next. If the current thing is three and four has appeared in the past, two is more likely to come next. But if you have multiple destination tokens they'll all want the same source token. For example, the phrase keep in mind can be a skip trigram. Really it should be a trigram. But tiny models aren't very good at figuring out what's exactly the previous position. Keep at bay is another trigram but in an at will both look at the same keep token and so they must boost boost both at and mind for both of them so it'll also predict keep in bay and keep at keep at mind and yeah and. Possibly we should move on to induction heads, which are a good illustrative example.

I was yeah, I.
SPEAKER: Tim Scarfe
Was going to come on to that. So on these induction heads you've said that they seem universal across all models. They underlie more complex behavior like few shot learning. They emerge in a phase transition and they're crucial for this in context learning. And you said that sometimes specific circuits underlie emergent phenomena. And, you know, we may

want to predict or understand emergence by studying these circuits. So what do we know so far?
SPEAKER: S4
A lot of questions in there.
SPEAKER: Neel Nanda
All right. All right. Taking this in order. So what is an induction head? I've already mentioned this briefly. Text often contains repeated subsequences like after Tim scarf may come next, but if Tim Scarf has appeared like five times then it's much more likely to come next in Toy two Layer attention only language models. We found this circuit called an induction head which does this. It's a real algorithm that works on, say, repeated random tokens and we have some mechanistic understanding of the basic form of it where there's two attention heads and two different layers working together. The the later one, called an induction head looks from Tim to previous occurrences of scarf. The first one is a previous token head which on each scarf looks at what came before and is like Ah, this is a scarf token which has Tim before and then the induction head looks at tokens where the token before them was Tim or where the token before them was equal to the current token. And when the attention induction head decided to look at scarf the which are determined purely by the QQ matrix, it then just copies that to the output which is purely done by the matrix. And I think induction heads are a really interesting circuit case study because induction heads are all of the interesting computation is being done by their attention pattern like Tim Scarf could be anywhere in the previous context and this algorithm will still work. And this is like important because this is what lets the model do tracking of long range dependencies in the text where it looks far back and you can't bake this in with a simple thing like convolutional layer. Um, in fact transformers seem notably better than old architectures like Lstms and RNNs in part because they have induction heads that let them track long range dependencies and yeah, and more generally it often is the case that especially late layer attention heads the bit is kind of boring, it's just copying. But figuring out where to look is where all of the interesting computation lies.
SPEAKER: Tim Scarfe
So so first of all, just to clarify because people will know what an attention head is, but an induction head is one of these circuits that that that you're talking about just so people understand. And we should get on to this relationship between induction heads and the emergence of In-context learning. And also you said it's very important that we have this scientific understanding, you know, with respect to studying emergence, but rather that than just framing of interpretability kind of makes better models.
SPEAKER: Neel Nanda
Yeah. So okay, so maybe I should first explain what emergence is.

So let's do that. Uh.
SPEAKER: Tim Scarfe
I'd be really, really interested if you could just give me the simplest possible explanation of what you think emergence is.

Sure.
SPEAKER: Neel Nanda
Emergence is when things happen suddenly during training and go from not being there to being there fairly rapidly in a non-convex way rather than gradually developing.
SPEAKER: Tim Scarfe

That's interesting you said that because I think of emergence as a surprising change in macroscopic phenomena and it's an observer relative term, which means it's it's always from the perspective of another scale.
SPEAKER: Neel Nanda
Hmm.
SPEAKER: Tim Scarfe
So just a transient change in in perplexity or capability or something in my mind wouldn't entail emergence.
SPEAKER: Neel Nanda
Like it would need to be some qualitative meaningful thing rather than just, oh, the loss curve got notably better in this bit.
SPEAKER: Tim Scarfe
I think it's definitely related to some notion of surprise which is inherently relative.

Um.
SPEAKER: Tim Scarfe
Yeah, let's not get hung up on that. So okay, it's let's say it's a transient change in something.
SPEAKER: Neel Nanda
Yeah. Uh, I wouldn't call it transient. It's like an unexpected sudden change, though Unexpected has so much semantic meaning on it that I don't want to use. But this is an infinite rabbit hole.

Yes, but.
SPEAKER: Tim Scarfe
I think the scale thing is relevant as well. So we are programing neural networks at the microscopic scale and there's some macroscopic change in capability. So it's some.

Yeah, Yeah.
SPEAKER: Neel Nanda
And there's like lots of different dimensions you can have emergence on. You can have it as you train a model on more data you can have as you make the models bigger. And these are both interestingly different kinds. One of the more famous examples is chain of thoughts and few-shot prompting where Gpt3 is pretty good at this. Earlier models were not good at this. This was kind of surprising. Chain of thought is particularly striking because you people just noticed a while after Gpt3 was public that if you tell it to think step by step it becomes much better. Um, there's this recent innovation of Tree of Thoughts that I'm not particularly familiar with but I understand is kind of like applying Monte Carlo tree search on top of chain of thoughts. Yes.

Yes.
SPEAKER: Neel Nanda
Where you're like, well there's many ways we could branch at each point. Let's use tree search algorithms to find the optimal way of doing this.

Yeah, but with.
SPEAKER: Tim Scarfe
Um, let's say scratch pad and chain of thought, I don't necessarily see that as an emergent well, maybe it is. Maybe there's an emergent reasoning capability that comes into play when

you have a certain threshold size model. But I think of it more as kind of having an intermediate augmented memory in the context. So you're kind of filling in a gap in cognition by saying you're allowed to It's not just remembering things, it's also reflecting on things that didn't work.
SPEAKER: Neel Nanda
Yes. So yeah, clarifying. When I say emergent, when I say chain of thought is an emergent property, I mean the capacity to productively do chain of thoughts is the emergent thing. And telling the model to think step by step is a user driven thing. Yeah, but I don't know, I kind of.

Just.
SPEAKER: Tim Scarfe
As a point of order though, was it just that it was discovered after after GPT three or would it work on GPT two?

Uh.
SPEAKER: Neel Nanda
I would have guessed it doesn't work very well on GPT two, but I've not checked. I'd be pretty interested. I'm sure someone has looked into this. I haven't looked very hard, I guess like so a lot of my motivation for this work comes from I care a lot about risk and alignment and how to make these systems good for the world. And when I see things like, Oh, we realized that you can make GPT three much better by asking it to think step by step. I'm like, Oh no. Um, what kinds of things could the systems you make be capable of that we just haven't noticed yet?
SPEAKER: Tim Scarfe
That's the concern that the the genie's already out of the bottle. And I mean, DeepMind just published this Tree of Thought paper. It's really simple idea. It's basically a star search over trajectories of prompts. And you use the model itself to evaluate the value of a trajectory. And I could have done that. Anyone could have similar thing with auto GPT and all this stuff. Um, I'm more skeptical than you are. I think in the case of Tree of Thought, it closes a capability gap in respect of certain tasks which were not working very well because they don't have that kind of system. Two models don't seem to plan ahead very well, but I still think that it's not just going to magically turn into superintelligent. I mean, we can talk about this a little bit later, but yeah, Okay.
SPEAKER: Neel Nanda
Yeah. So yeah, I think this is also pretty relevant to like much more near-term risks Like yeah, I don't know. There's lots of things that are sufficiently capable model could do that might be pretty destabilizing to society like right actually much better propaganda than human writers can or something and if Tree of Thought makes it possible to do that in a way that we did not think was possible when GPT four was deployed, that's like an interesting thing that I care about noticing. There's not a very good example, but.
SPEAKER: Tim Scarfe
Yeah, it is.

Um.
SPEAKER: Tim Scarfe
But being able to I mean, first of all, it's been possible to create misinformation for a long time.
SPEAKER: Neel Nanda

This is why I specified be able to do it notably better than humans can. I totally agree that like doing it a bit more cheaply and a bit more scale doesn't seem obviously that important. You could argue that like, I don't know, being being a spam bot that. Feels indistinguishable from a human is like a more novel thing that's actually different. Yeah, but I know this is like an off the cuff example. I don't want to get too deep into this because it's not a point I care that deeply about.

Yeah, I mean.
SPEAKER: Tim Scarfe
We can come back to it in a bit, but I think we are nearly already.

There. Yeah.
SPEAKER: Tim Scarfe
So this irreversibility thing, we we don't know. Uh, computer games are photorealistic. Chat bots are indistinguishable and art is pretty much indistinguishable. And that could work. I mean, I spoke to Daniel Dennett about it last week and he said he's really worried about the epistemic erosion of our society. More so interestingly than the ontological erosion. And I discovered later that's because he's not a big fan of anything ontological. But um, yeah, it's it is potentially a problem, but I guess to me people might overestimate the scale and magnitude of change of this. I feel that I know I don't want to echo Sam Altman here, but he said that we are reasonably smart people and you know, we can we can adapt and recognize, you know, deepfakes and so on.
SPEAKER: Neel Nanda
But yeah, yeah. He's a complicated societal questions. I guess I mostly just have the position of man, it sure is kind of concerning that we have these systems that could potentially pose risks, but you don't know what they do and decide to deploy them and then we discover things they can do. And I think that the research direction I'm trying to advocate for here is just better learn how to predict this stuff more than anything, which hopefully we can all agree is like an interesting direction and there's all kind of debates about is emergent phenomena like actually a real thing like this recent is this a mirage paper which I think was a bit overclaiming but does make a good point that if you choose your metric to be sufficiently sharp, everything looks dramatic. Um, one thing I've definitely observed is if you have an accuracy graph with a log scale x axis for Grokking, it looks fantastically dramatic and I was very careful to not do this in my paper because it is cheating. Um, but yeah, so my particular hot take is that I believe emergence is often underlain by the model learning some specific circuit or some small family of circuits in a fairly sudden phase transition that enables this overall emergent thing. And this paper led by Katherine Olson In-context learning in induction heads is a big motivator of my belief of this. So the idea of the paper is we have this we found induction heads in these toy models. We somewhat mechanistically understood them, at least in the simplest case of induction. We use this to come up with more of a behavioral test for whether an induction heads you just give them all a repeated random tokens and you look at whether it looks induction y and we found that these occurred in basically all models we looked at up to 13 B even though we didn't fully reverse engineer them there. And we then found that this was really deeply linked to the emergence of In-context learning a lot of jargon in there. So let's unpack that in context. Learning already briefly mentioned it's like tracking long range dependencies in text like you can use what was on, which was three pages ago to predict what comes next in the current book, which is a non-trivial thing. It is not obvious to me how I would program a model to do in

context. Learning is emergent if you operationalize it as average loss on the 500th token versus average loss on the 50th token, there's a fairly sudden period in training where it goes from not very good at it to very good at it.
SPEAKER: Tim Scarfe
Just a tiny point of order there. One interesting thing about In-context learning is you're learning at inference time, not training time, but you're not changing anything in the underlying model, which means anything it can do presumably must be materializing a competence which was acquired during training. So it's coming back to this periodic table thing, right? So it's learned all these platonic primitives. You do this in context learning. You say, I want you to do this. Here's an example and it kind of you know, you've got all of these freeze dried periodic computational circuits and they spring into life and they compose together and they do the thing.

Yes.
SPEAKER: Neel Nanda
Yes. Yeah. I think induction heads are, to my eyes the canonical example of an inference time algorithm stored in the model's weights that gets applied. And I'm sure there's a bunch more that no one has yet found. And yeah, a lot of my modelers that prompt engineering is just telling the model which of its circuits to activate and just engaging with various quirks of training that have made it more or less terrible in different ways. And yeah, so induction heads also emerge in a fairly sudden phase transition. And we and exactly the same time and we present a bunch more evidence in the paper that there's actually a causal link here. Like one layer models have neither the in-context learning or the induction heads phase chain because they can't do induction heads because they're only one layer. And why? But if you adapt the architecture so they can form induction heads with only one layer, now they have both of these phenomena. If you beat induction heads, in-context learning gets systematically worse. And a particularly fun qualitative study was looking at soft induction heads. Heads that seemed to be doing something induction in other domains like a head which attends from the current word in English to the thing after the current word in French or more excitingly, a few shot learning head on this random synthetic pattern recognition task we made where it attended back to the most relevant examples to the current to the current one. And my interpretation of all this is that there's something fairly fundamental about the induction algorithm for In-context learning. So the way I think about it, let's say you've got to you want to learn some relation, you've got some local context A and some past context B and if you observe A and you observe B in the past, this gives you some information about what comes next. Um, there's two ways this could work out. It could be symmetric. B helps A and A helps B or asymmetric B helps A, but A does not help B if they're the other way around. Asymmetric might be like knowing the title of a book tells you what comes next but knowing what's in a random paragraph in the previous bit doesn't tell you the title. Um. While symmetric is like a no English sentence helps French sentence, French sentence helps English sentence. And if you have like n symmetric relations like English, French, German, Dutch, Latin, whatever where each of them helps each other, this is really efficient to represent because rather than needing to represent n squared different relations separately like you would in the asymmetric case, you can just map everything to the same latent space and look for matches. And fundamentally this is what induction heads are doing. They're mapping current token and previous token a thing in the past to the same latent space and looking for matches. And to me this is just like a fairly natural primitive of attention. And this is exciting because a we found this deep primitive by looking at Toy Tulare attentional

models. B It was important for understanding and ideally for predicting the emergent phenomena of in-context learning. And my two takeaways I have from this about work we should be doing. The first is we should be going harder at looking at toy language models like open sourced a scan of 12 of them and I'd love to see what people can find in one layer models with MLPs because we really suck at transformer layers and one layer should just be easier than other ones. And the second thing is I really want a better and more scientific understanding of emergence. Why does that happen? Really understanding particularly notable case studies of it testing the hypothesis that it is driven by specific kinds of circuits like induction heads or at least specific families of circuits, even though I don't know, you could argue that because we haven't fully reverse engineered the things in the larger models we really know it's actually an induction head. And yeah, more generally, a lot of my vision for why Macintalk matters is this kind of scientific understanding of models like don't care about making models better but care about knowing what's going to happen, knowing why stuff happens, achieving real understanding and getting a scientific understanding of things like emergence seems like one of the things Mektup might be uniquely suited to do but also known checked very hard. And you, dear listener, can be the person who checks.
SPEAKER: Tim Scarfe
So there was a paper by Kevin Wang et al. Called Interpretability in the Wild, a circuit for indirect object identification in GPT two small which found a circuit for indirect object identification. So they discovered back up named mover heads, which normally don't do much. They take over when the main name mover, head or ablated and they said mechanistic interpretability as a validation set for more scalability techniques. They've understood a clear place that these ablations can be misleading. So.
SPEAKER: S5
Yeah.
SPEAKER: Neel Nanda
So yeah, bunch one pack in there. So I really like the interoperability in the wild paper. Also, Kevin was only 17 when he wrote it and like, man, I was doing nothing remotely as interesting when I was in high school, so props to him. Um, but also a sign of how easy it is to pick low hanging fruit and do groundbreaking interpretability work. Um, such a young field. Um, I know. It's so impressive.

Yeah.
SPEAKER: Tim Scarfe
I've just checked his Twitter.

Hey, Kevin.
SPEAKER: Neel Nanda
And yeah, so to me, the underlying. Yeah. So zooming out a bit, I think there's a family of techniques around causal interventions and their use in mechanism. McIntyre That's useful to understand here. So the core technique is this idea of activation patching where so let's so one of the problems with understanding a model's features and circuits is models are full of many, many different circuits. Each circuit does not activate on many inputs, but each circuit will activate, but on each input many circuits will activate. And in order to do good McIntyre work, you need to be incredibly surgical and precise, which means you need to learn how to isolate a specific circuit. And let's consider a statement like, um. The Eiffel Tower is in Paris versus the Colosseum is in Rome. These are both there's lots of features happening. There's lots of circuits being activated on the Eiffel Tower in Paris. This is an English you're

doing factual recall. You are outputting a location. You are outputting a proper noun. This is a European landmark. ET cetera. ET cetera. And like I want to know how the model knows the Eiffel Tower is in Paris. But the Colosseum is in Rome. Controls almost everything apart from the fact. And so what I can try to do is causally intervene on the Colosseum, run and replace, say, the output of an attention head with its output on the Eiffel Tower prompts and see how much this changes the answer from Rome to Paris. And this Yeah, this patch can let me really isolate how the circuitry for just this specific thing works and there's all kinds of work around this. Obnoxiously all of it uses different notation like resample ablations and causal tracing and causal mediation analysis and interchange interventions. All similar words, basically the same thing. Um, but yeah, um, the really key insight here is this kind of surgical intervention. A classic technique in interpretability is ablations where you just set something to zero and it's kind of janky because if you break something in the model which wasn't interestingly used for the task, then everything dies. Or if you break it in interesting ways, everything dies. For example, in GPT two small, almost every single task breaks If you delete the zeroth layer. Um, yeah. As far as I can tell, the zeroth multi-layer is kind of an extended embedding. Um gpt2 small has tied embeddings and embeddings so their transpose of each other which is wildly unprincipled in my opinion and the model seems to be both using this for just tokenization and combining nearby things with the first attention layer zeroth attention layer and just undoing the tightness. Um, but this means that basically everything is reading from that and I've seen people do zero ablations and everything and be like, Oh, this is an important part of the circuit. Let's get really sidetracked by this. Um, because the effect size is so big. Yeah. Oh man. Being a McInturff research fills my mind with such bizarre trivia like this. It's great models. So bizarre. Um, and so, yeah, um, this causal intervention, there's kind of two conceptually different kinds of interventions. You can take the Eiffel Tower prompt patch in something from the Colosseum and see if it breaks the ability to output Paris to verify which bits kind of are necessary such that getting rid of them will break something. Or you can patch something from the Paris, run into the Colosseum, run and see if that makes it output Paris, which is testing for stuff that's sufficient. I call the first one a re sample ablation because you're messing up a component by re sampling. And the second one denoising or causal tracing because you're intervening with like a bunch with like a bit of information and seeing if that is sufficient for everything else though none of these names are good. I would love someone to come up with better names and there's all kinds of families of work building on this. Like I have this post called Attribution Patching that tries to apply this at industrial scale by using gradients to approximate it, which is fast enough that you could take GPT three and it's 4 million neurons and do attribution matching on all neurons at once on every position. Uh, great, great post. Redwood Research has this technique called causal scrubbing, which I view as activation patching gone incredibly hard and rigorous that tries to come up with an automated metric for saying this hypothesis about a model is actually accurate for how it works. Um, where it's kind of complicated, but the core idea is you think of a hypothesis as saying which resample ablations are allowed and you make all of the resample ablations that should be allowed like these components of the model shouldn't really matter. So we can just patch in stuff from random other inputs. Um, if you've got. An induction head. You might think the induction head cares about the current token and the thing before the previous the thing before the past token that it's going to induct is going to inductively attend to. So let's replace the token that it's going to be attending to with a token from a different input but with the same token before it. My hypothesis about the induction head says this should be allowed. So let's do that.
SPEAKER: Tim Scarfe

I wouldn't want to induce a rant, but the the metric you use is really important, right?
SPEAKER: Neel Nanda
Yes, this is one of my hobbyhorses. So, um, some of the original work looking at the patching stuff like David Bao and Kevin Meng's excellent Rome paper uses the probability of Paris as their metric. And there are other papers that use things like accuracy as their metric. And generally I think of metrics as being on a spectrum from like soft to sharp. So generally I think of models as thinking in log space. They are kind of acting like Bayesians. They are trying to figure out if something's in Paris and there'll be five separate heads that each contribute one to the correct logits And each of these can be thought of as like one bit of information and together they get you the right probability of say 0.8. But if you patch in each one in isolation, the probability changes negligibly because probability is exponential in the log logits. So using probability you're like, Oh this, this head patch doesn't really matter. And so in this paper they did this thing of patching in like ten adjacent layers at once. And to me a really cool principle of this kind of causal intervention and mechanistic technique is you want to be as surgical as possible, to be as deeply faithful as possible to what the neural model is actually doing.

So in this.
SPEAKER: Tim Scarfe
Case there was an interaction between them. They were effectively making several interactions or interventions at once.
SPEAKER: Neel Nanda
Yes. Yeah, they were like replacing ten adjacent layers and no patching things in different layers. There's always a bit weird. I don't think that part's that objectionable. I mostly just feel like if you choose a metric like log prob it allows you to be much more surgical about how you intervene. It allows you to identify subtle effects of things. Accuracy is even worse because accuracy is basically rounding things to 0 or 1. So like if the threshold is 2.5, any individual patch does nothing, any resample ablation does nothing. But if you patch in like the ten adjacent layers it will do everything and this can be kind of misleading. Another one I often see people do is um, they're. Trying. They look at things like the the rank of an output. Like at which point does the model realize Paris is the most likely? Next token and this can be super misleading because this will make you think the third head is the only head that matters when really all five of them matter. The order is kind of arbitrary and yeah, I've seen papers that I think got somewhat misled by using metrics like this. And metrics they matter so much. It's so easy to trick yourself. My high level pitch is just McInturff is great. McInturff is beautiful. Also, the field is incredibly young. There's maybe 30 full time people working on it in the world. There's a ton of low hanging fruit. I've done major research in this field and I've been in it for like less than two years. I would love people to come and help and help us solve problems and do research here and we'll link to my post on getting started and my sequence called 200 Concrete Open Problems in the description to this hopefully. And of course yeah, I think that's just it's not that hard to get started. It's really fun. Hopefully I've nerd sniped you with at least one thing in this podcast and if you're at least vaguely curious, it's just really easy to open one of the tutorials linked in my posts and just start screwing around and I'd love to see where you can find.
SPEAKER: Tim Scarfe
Beautiful.
SPEAKER: Neel Nanda

Also, the DeepMind alignment team is currently hiring and people should apply. Which includes hiring for our mechanistic interpretability team.
SPEAKER: Tim Scarfe
Amazing. Do they have to do leetcode?
SPEAKER: Neel Nanda
I have no idea. Can't remember.

Yeah.
SPEAKER: Tim Scarfe
Yeah. We, um. We did an amazing video with, uh, Petr Veljkovic. Um, I gave him one of my leetcode challenges, and annoyingly, he aced it.
SPEAKER: Neel Nanda
Oh, I'm so sorry.
SPEAKER: Tim Scarfe
All of that. It's all that DeepMind interviewing interview practice. Anyway, okay, let's talk about superintelligence now. I spoke with our mutual friend Robert Miles about a month ago.

Rob So.
SPEAKER: Tim Scarfe
Great. He's he's a lovely chap. Spoke all about alignment and he accused me of over philosophizing everything because I was talking all about intelligence, one of my favorite topics. And he said, Well, what about fire? Fire is something that people didn't understand millennia ago, but they knew that it burnt and they knew that it was bad and this is like this is like a fire, which is very interesting. And maybe we can bring in a little bit of effective altruism as well. So, um, you know, I interject. Please do, please.
SPEAKER: Neel Nanda
If there is one thing I have learned from the past decade of machine learning progress is that you do not need to understand a thing in order to make it. And this extends to things that are smarter than us and which are capable of leading to catastrophic risks.
SPEAKER: Tim Scarfe
Yes. Yes. Well, let let's, um. I'll step back a tiny bit and then we'll and then we'll get there because there's the hypothetical nature which I guess I have a bit of a problem with. Now, about ten years ago I was one of the first supporters of Sam Harris's podcast and he's quite aligned to EE and he was talking about this very noble idea that everyone matters equally and people on the left should get on board with that intrinsically. And this idea that we should quantitatively analyze the impact of charity work and solve an optimization problem and earning to give. And a lot of the stuff that MacAskill spoke about and also philosophers like Peter Singer and the focus seemed to be primarily on alleviating poverty, which we and we we don't say the biggest problem, we say a problem. This is another thing our friend Robert Miles said. You said the problem is when people talk about the problem, there can be more than one problem. But anyway, so it's a big it's a big problem. And um, recently you and I can agree that circles have really laser focused in on existential risk from AI as opposed to other more plausible risk concerns like pandemics or even nuclear war. I'm not not to say that they don't focus on that, but.

I am.
SPEAKER: Neel Nanda
Going to push back on other more plausible risks.

Go on.
SPEAKER: Neel Nanda
Go on. I just wanted to register an objection. Feel free to go register.

Okay.
SPEAKER: Tim Scarfe
So and you know, cynically from from my point of view, I see I see the influence of Eliezer Bostrom Hanson. ET cetera. Kind of shifting the focus onto X risk. And part of part of the reason for that is also this kind of overly intellectual focus on long termism. And it's done in a very intellectualized way. So it's based on the utility function now incorporating future simulated humans on different planets, you know, a long time away in the future and making all of these intellectual jumps. So let's let's start there.

What's your take?
SPEAKER: Neel Nanda
So much stuff to respond to in there. Good. So all right. A couple of things. The first so cards on the table. Um, I care a lot about existential risk. Yes. The reason I work on mechanistic interpretability is because I think that understanding the mysterious black boxes that are potentially smarter than us and may want things wildly different than what we wanted them to want is just clearly better than not understanding them. Yes, and I think mechanistic interpretability is a promising path here. So and I also would consider myself an effective altruist and a rationalist. So cards on the table. There's my biases. Um, so I generally think it's more productive to discuss is AI catastrophic and existential risk a big deal than is it the biggest deal or is it worth more resources on the margin than global poverty or climate change or ethics? And like there's just lots of problems. I care way more about convincing people that could be in your top ten than it should be in your top one because I feel like for most people it's not in their top 1000. And there's just so much divisiveness between, say, the ethics community and the alignment community about who's problem is a bigger deal and like both are big problems. Why are we arguing and.

What part of.
SPEAKER: Tim Scarfe
This is about our moral intuitions? And this is something I spoke a lot with Connor about. You know, he said that in many ways he's got this technical empathy. So sensory empathy is I really care about my family there these concentric circles of moral status. I really care about my family. And if I try really hard, I can care about people in other countries and so on. And then if I try really, really hard, I can care about future simulated lives on Mars. And Connor said the idea of this movement is about galaxy branding yourself into being the most empathetic person imaginable. But it's a kind of empathy that people don't understand.
SPEAKER: Neel Nanda
Um, yeah. So okay, so a separate bit of beef I have is with the entire notion of Longtermism right? So longtermism is this idea. Okay. So Longtermism is generally caring about the long term future. Yeah. There's like the strong form of value in the future basically entirely dominates things today and weaker forms of just this really, really matters. And a common misconception about risk and safety is that you should only work on this if you are a long termist that you know it's a 1 in 1,000,000,000 chance of mattering. But there's a quintillion future lives. This outweighs everyone alive today in moral worth or well, we're only going to

get AGI in like 500 years, but we're going to work on it now just in case. And like I think both of these are just nonsense. Um, like I guess there's a concrete example. Um, effective altruists have worked on pandemic prevention for many years and I think it was just clearly the case that pandemics are a major threat to people alive today. And I like to feel that we've been proven right.

No one's.
SPEAKER: Tim Scarfe
Going to argue at that.
ç
Point.
SPEAKER: Neel Nanda
And you know, everyone's being like, but altruist, why are you working on safety? This obviously doesn't matter. And you know, I feel like we got one thing right. Um, can.

I can I be.
SPEAKER: Tim Scarfe
Really skeptical, though, for a second? Because, I mean, you're working for DeepMind. There's so much prestige and money attached to risk. Elon Musk is talking about it all the time. Whereas you could be a scientist working on pandemic response responses. And I mean, let's be honest, it wouldn't be anywhere near the same level of prestige.
SPEAKER: Neel Nanda
Yeah. So. Couple of takes it. Definitely is the case that I a good chunk of why I personally am working on x-risk rather than say bio risk is that I am a smart mathematician. I like I like makin up. I do not think I'd be good at biology in the same way and I also would personally assert that X risk is more important and like more pressing. But you know, I'm biased and I think it's fair to flag that bias. Um, in terms of prestige. So I've only really been working on this stuff properly for the past two and a half years, which is I mean it's changed dramatically like in the last six months we've gone from well, we've really ever going to get AGI to Oh my God, GPT four exists. Geoffrey Hinton has left Google to loudly advocate for X-risk. Yoshua Bengio is now loudly advocating for X-risk. It's two thirds of the Turing winners for deep learning.

You'll never get the third one.
SPEAKER: Neel Nanda
Yeah, we're never going to get the third one. Yann LeCun has made his position very, very clear. Yeah. Um, but you know, it's a majority. I'll take it.

Yes. And or the fourth one? Yeah. Yeah.
SPEAKER: Tim Scarfe
Um, he's coming on our podcast, actually.
SPEAKER: Neel Nanda
Uh oh. Who was the fourth one?

Schmidhuber. Ah, yeah.
SPEAKER: Neel Nanda
Yeah. That seems hard. I'm very curious to hear the Schmidt. Who? Your episode.
SPEAKER: Tim Scarfe

Oh, yeah. He's even more virulently against than Yann. I'm afraid to say so. Two out of two.

Two out of four.
SPEAKER: Neel Nanda
I'm interested to hear anyway, so. Yeah. Um, and yeah, in terms of prestige and no, I gather that say seven years ago it was basically just not it would be like pretty bad for your career. You would not be taken seriously if you mentioned caring about risk. Your papers would be rejected. I hear a story of Stuart Russell at one point talked to a grad student of his about how Stuart was concerned about risk. The grad student was also really concerned and freaking out, but they've been working together for years and neither had felt comfortable mentioning it and a lot of people who are still in the field were doing this stuff then, which makes me somewhat reject the prestige argument, at least for senior people in the field.
SPEAKER: Tim Scarfe
I think there's a difference with Stuart Russell in particular. He's very credible and he and.

I'm not.
SPEAKER: Tim Scarfe
Oh, I didn't mean I didn't mean you. I was talking I was talking about the two godfather because the thing that. Maybe I shouldn't say this, but I was surprised that Bengio and Hinton came out in the way they did. And I the reason I didn't like what they said was I felt that they were implying that current AI technology could pose an existential threat. And what I'm getting from you and what I'm getting from Russell is and also from Robert Miles is that this is a very real potential threat in the future. But it's not a current threat.
SPEAKER: Neel Nanda
Yes, very real potential threat in the future, though I hesitate to confidently assert say this will not be a threat in the next five years or something. It's like pretty hard to say. Interesting. I'm not confident. I agree with your assessment of Bengio and Hinton though they've spoken a bunch publicly, so I'll defer if you can point to specific writings. But for example, Bengio signed the pause for six months more powerful than GPT four letter and I don't know, I don't think the letter was asserting that the letter definitely wasn't asserting four was an existential risk. It wasn't confidently asserting GPT five would be. It's being like, Yeah, we need more time and slow down and caution.
SPEAKER: Tim Scarfe
Yeah, maybe I'm reading too much into that, but it seemed to me that I mean, Hinton said that ChatGPT now contains all of the world's knowledge and this chat bot knows everything and it could potentially do very harmful things. And I interpreted it possibly incorrectly that they were talking about reasonably current or next generation risks.
SPEAKER: Neel Nanda
For I mean I can't talk for them. I also don't know there are lots of near-term risks. There's long term risks. I consider it my job to think hard about the long term risks and try to guard against those. And I think lots of other people jobs is to focus on like the near-term risks and both are like great forms of work. I don't know. One reason I like interpretability is I think it is just broadly useful across all of them. So what I consider to be my job might just not even matter. Yes. But yeah. Um. Yeah. No I probably will not do not want to get deeply into interpreting what other people have said. Um I.

Yeah. Well, could.
SPEAKER: Tim Scarfe

I could I ping you just a couple of quick questions. So first of all, you know, there's this idea of negative utilitarianism. I mean, do you think minimizing suffering is more important than maximizing happiness?
SPEAKER: Neel Nanda
Mhm. Nah.

No, not sure I've.
SPEAKER: Neel Nanda
Got a more deep answer than that. I mostly think a lot of this intuitive reasoning is more driven by intuition than anything else.

But it's a bit like this.
SPEAKER: Tim Scarfe
Matrix thing we were talking about, you know, which is that metrics if you want to have. Would you like to tolerate some spiky necks for some average happiness?
SPEAKER: Neel Nanda
Yeah. So I know I have like a general frustration with these discussions getting too philosophical. Um, this is I'll do my big issue when I hang out with effective altruists who really love moral philosophy and population ethics. Yes. I don't know. I have this forum post called Simplify Pitches to Holy shit risk. I'm just like so I don't know. Um, if you actually look at some of the concrete work people tried doing on things like timelines and risk, there's this report from a Jakarta at Open Philanthropy that gives a 30 year median timelines to to AI that's transformative which she since updated to 20 years there's a report by Joseph Carlsmith that estimates about a ten ish percent chance of a major catastrophe from this. Yeah and if you just take those numbers, this is clearly enough to reach pretty high on my list of concerns for people alive today.

Okay.
SPEAKER: Tim Scarfe
Okay.
SPEAKER: Neel Nanda
And I think these are bold, empirical claims and I think it's great to debate them in the empirical domain. But to me, this doesn't feel like a moral question. It just feels like from common sense assumptions, if you believe these empirical claims, this stuff is a really big deal.

Okay. Okay.
SPEAKER: Tim Scarfe
Let's let's take another couple of steps. So first of all, we save this till later. Um, I think deception is very important. And Daniel Dennett, when I spoke with him, he uses this notion called the intentional stance, which basically means that if you use a projection of purposes, goals, agency, etcetera, in order to understand the behavior of an agent, possibly a simulated agent, then for all intents and purposes it has agency, it can make decisions, it has moral status, it has lots of different things like that. And he would say that without an intentional stance, without agency, it's impossible for a model to lie or deceive us. Now, what do you think would be the bar for something like a GPT model to deceive us and why?
SPEAKER: Neel Nanda

Yeah, so. Before I give takes, I will generally reinforce Rob's vibe of, well, if you have no idea how fire works but you know that it burns you, That's kind of the important thing. Like maybe a model has just this random learned adaptation to output things that are designed to get a user to feel and believe a certain way that isn't intentional and isn't deceptive in some true cog sci sense. But it's like enough for this to be a big deal that we should care a lot about.

Okay, okay with that.
SPEAKER: Neel Nanda
With that aside, yeah. Um, yeah. So I'm definitely hesitant to ascribe an overly confident view of what's going on here. Um, and I think lots of early discourse and alignment around things like utility maximization and around things like. These things are just paperclip maximizers, etcetera is kind of misleading and I don't think it is an accurate model of how GPT seven F plus plus plus is going to work or that's my prediction. One thing that is pretty striking to me is I just feel like we're pretty confused on both sides of this. Like I do not feel like I can confidently claim that these models will demonstrate anything remotely like goals or intentions, but it also don't feel like you can confidently claim that they won't. And I'm not talking like 99.99% confidence. I'm talking like 95% plus confidence either way. And one of my visions for what being good at Macintalk might look like is being able to actually get grounding for these questions because I think ultimately these are mechanistic questions. Behavioral interventions are not enough to answer like does this thing have a goal in any meaningful sense? But yeah, my like very rough soft definition would be is the model capable of forming and executing long term plans towards some goal potentially if explicitly prompted to like auto GPT or just spontaneously is it capable of actually carrying out these plans and does it form and execute plans towards some objective that is like encoded in the model somewhere? Um, and I don't know, I think it's pretty plausible that the first dangerous thing is like chaos. GPT seven where someone tells it to do something dangerous and it gets misused more so than it's like misaligned. And I care deeply about both of these risks.

Okay.
SPEAKER: Neel Nanda
So yeah, first one is more of a governance question than a technical question and thus is less where I feel like I can add value.
SPEAKER: Tim Scarfe
So I agree with you on all of that. So yeah, being less confused about what's going on inside the models and.

Some great, you.
SPEAKER: Tim Scarfe
Know, using interpretability to figure out whether they actually do have agency or goals and sometimes they do the right things for the wrong reasons. Auditing models that seem aligned before they're deployed is something that you've told me before.

That's so great.
SPEAKER: Tim Scarfe
And um, you know, just being able to check more deeply that it truly is aligned. But I wanted to talk a little bit about, um, this interesting paper from Katja Grice. So she wrote a response called It was on Lesswrong Debunking the AI Apocalypse a comprehensive analysis of counterarguments to the basic risk Case X Risk. And the reason I read it is so many of the

comments were destroying me and Doug after we interviewed Rob and they said, Well, if you're going to criticize X risk, I mean at least go and read Kathy Grace's response. So I did. So I did. Here we go. So she she basically made two big counterarguments that intelligence might not actually be a huge advantage. And about the speed of of growth is ambiguous. But I first want to touch on what you said before, which is about this notion of goal Directedness So alignment people say that if superhuman AI systems are built, any given system is likely to be goal directed and the Orthogonality thesis and instrumental goals are cited as aggravating factors and the goal directed behavior is likely to be valuable. So economically goal directed entities may tend to arise from machine learning training processes, not intending to create them, which is kind of talking about some of the emergent behaviors that we were talking about earlier with respect to Othello, for example. And coherence arguments may imply that systems with goal directedness will become more strongly goal directed over time, which is apparently something that is argued for. So I'm thinking what does goal even mean? I mean we anthropomorphize abstract human intelligible concepts like goals and they they really are emergent because they emerge from these low level interactions in the cells in your body and then you get these things that we recognize to be goals observer relative as we were talking about before but they're just graduated phenomena from smaller things, right? So what does it even mean to have a goal?
SPEAKER: Neel Nanda
Yeah. So. Couple of thoughts on that. Again, you ask questions with a lot of content in them. No problem.
SPEAKER: Tim Scarfe
I can only.

Apologize, but.
SPEAKER: Neel Nanda
I mean, as someone who accidentally writes 19,000 blog posts, 19,000 word blog posts all the time. Relate. Um, anyway, so what am I saying? Um, so the way.
SPEAKER: Tim Scarfe
Yeah, it's a vague.

Concept, right?
SPEAKER: Neel Nanda
Yeah. So I definitely want to try to take. So there's the mechanistic definition of the model forms plans and it evaluates the plans according to some criteria or objective and it executes the plans that score better on this. And I would love if we get to a point where we can look inside a model and look for circuitry that could be behind this or not, that would feel like a big milestone for me. On Wow, I really believe Mac and Apple Matter for reducing catastrophic risk from AI. Um, a second thing is that um yeah the kind of more behavioral thing of the model systematically takes actions that pushes the world towards a certain state. And I don't want I think there's a common problem in alignment arguments where people get too precise and too specific in a way that lots of people reasonably object to and a way which not necessary for the argument. Um, there's a really great paper called The Alignment Problem from a Deep Learning Perspective by Richard No, Laurence Tannen Saurine Mindermann. And this is probably my biggest recommendation for the listening audience of what I think is like a pretty well presented case for alignment and I generally pretty pro trying to make the minimal necessary assumptions. So for me it's kind of like some soft form of goal directedness of take actions that push the world towards a certain state. And another

important thing is there are a bunch of theoretical arguments for why goals would spontaneously emerge. Um, ideas around inner misalignment from work led by Evan Hubinger ideas around this coherence theorems and things like that which no, I find like a bit convincing, not that convincing, but then there's things will have goals because we try to give them goals and I'm like, Yeah, that's probably going to happen. Um, it's just clearly useful if you have a if you want to have an AI CEO or an AI helping run logistics or military operations to have something that's capable of forming and executing long term plans towards some objective. And if you believe this is what's going to happen, then the key question is are we capable of ensuring those goals are exactly the goals we would like them to be? And my answer for any question of the form, can we precisely make sure the system is doing exactly X machine learning is God? No, we are not remotely good enough to achieve this with our current level of alignment and steering techniques And to me this is like a more interesting point where it's not quite a crux for me, but it just seems like a lot easier to argue about Will people do this?
SPEAKER: Tim Scarfe
Yeah, it's interesting. I mean, Katia herself said that it's, um, it's unclear That goal directedness is favored by economic pressure training dynamics or coherence arguments. You know, whether those are the same thing as kind of goal directedness That implies a zealous drive to control the universe. And look at South Korea. They have goals and those goals I don't really subscribe to the to the dictator view of society. I assume they are somehow emergent. Yes. And similarly.

Sorry, South Korea or North Korea?
SPEAKER: Tim Scarfe
Sorry, North Korea. Did I say South Korea? I meant I.

Meant North Korea. Very different careers, Different different.
SPEAKER: Tim Scarfe
Goals, different goals. But um, but but you can think about goals in an AI system as either being ones which emerge from some low level or ones which are explicitly coded by us or ones which are instrumental. Right? And these are all a whole bunch of goals. Yeah. But we can't really control those. We can add pressures. How do we control what North Korea does?
SPEAKER: Neel Nanda
Uh, that sure is a question I'd love for someone to answer. Um, I don't know. Like I can give speculation. There's like there's the question of in practice, what do people do? Which is basically. The reinforcement learning from human feedback. And I expect people would apply that in this situation as well. I definitely do not believe we would be able to explicitly encode a goal in the system. Moreover, even if you can encode even if you could give some like scoring function, like make the score in this game high, this is not give you a model that intrinsically cares about that in the same way that I don't know evolution optimizes inclusive genetic fitness Don't give a fuck about inclusive genetic fitness Even though I care about a bunch of things evolution got me to care about within that. Like tasty foods and surviving. Um. Yeah. So we don't know how to put goals into systems. I basically just assert that we are not currently capable of putting goals into systems well and this is one of the main things the field of element thinks about and we're not very good at it and it'd be great if we were better at it. Um, in terms of yeah, I definitely don't want to make strong claims about to be dangerous. The goals need to be coherent or the goals need to. There needs to be like a

singular goal. Like I don't have a singular goal. Um, it's not obvious to me how these systems will turn out if they don't in any meaningful sense want a coherent thing then I'm a fair bit less concerned though. Well, I mean, there's many, many ways that human level AI would be good for the world or bad for the world or just wildly destabilizing and high variance of which Miss Lemon risk is one of them. And lots of the other ones would still apply like misuse and systemic risks. But leaving those aside. Um, yeah. I think if a model is just roughly pushing in a goal directed direction with a bunch of caveats and uncertainties and flip flopping, that still seems like a pretty big deal to me.
SPEAKER: Tim Scarfe
Okay.

Okay.
SPEAKER: Tim Scarfe
Katia, let's just cover her two main arguments. So she said that intelligence might not actually be a huge advantage. So looking at the world, um, intuitively big discrepancies in power are not to do with intelligence. And she said IQ humans you know, humans with an IQ of 130 earn roughly 6000 to 18,000 a year dollars more than average IQ humans. Elected representatives are apparently slightly more smarter, slightly smarter on average, but not a radical difference. Mensa isn't a major force in the world, and if we look at people who evidently have good cognitive abilities given their intellectual output, their personal lives are not obviously drastically more successful and anecdotally. So is it that much of a big deal?
SPEAKER: Neel Nanda
Yeah. So I think this is like a fair point. If we looked in the world and IQ or whatever metric of intelligence you want to use, um, clearly dramatically correlated with everything good about someone. Mean IQ correlates with like basically everything you might value in someone's life because we live in an unfair world but not dramatically. Um yeah. So I think this is a valid argument. I generally don't think you should model human level AI as like AI or like slightly superhuman. AI is like an IQ 200 human like for example, GPT four would argue knows most facts on the internet or many facts. Um and yeah knows many facts and this seems. Um. Jeopardy for knows many facts and this is sure an advantage over me. Jeopardy for knows how to write a lot of code and it knows how to take software and do penetration testing on it. It knows lots of social conventions and cultural things and has lots of experience reading various kinds of text written to be manipulative or manuals on how to make nuclear weapons. So I'm mostly going I'm going to hot on the knowledge point that just lots of different axes you can be human level or better in with which knowledge is one intelligence and reasoning is one Social manipulation abilities is another. Charisma and persuasion is another. I think these two are particularly important ones. Um, there's. Forming coherent plans. There's just like the ability to execute on stuff 24 over seven, running thousands of copies of yourself in parallel distributed across the world. Uh, there's running faster than humans and there's just like, lots of dimensions here. I think the IQ 200 human frame is helpful in some ways, but unhelpful in other ways, especially if it summons the like, nerdy scientist with no social skills whose life is a mess. Archetype. I say a nerdy scientist with no social skills. His life is a mess.
SPEAKER: Tim Scarfe
Okay. Yeah. I mean, this is this is the thing is, um. Because Rob said the same thing on on chess, it's possible for someone to be literally 20 times better than you. There's a huge dynamic range of skill and that's something we've not really seen in human intelligence. And it might be because of the way we measure it. It's possible that the way we measure it

doesn't even capture people with with with, you know, broader or better abilities. Let's just cover her last point quickly. So this is that the speed of intelligence growth is ambiguous. So this idea that I would be able to rapidly destroy the world seems prima facie unlikely to Katia since no other entity has ever done that. And she goes on. So the two common broad arguments is that there'll be a feedback loop in which intelligent AI makes more intelligent repeatedly until AI is very, very intelligent. Number two small differences in brains seem to correspond to very large differences in performance based on observing humans and other apes. Thus, you know, any movement past human level would take us to unimaginably superhuman level. And the basic counterarguments to that is that, you know, the feedback loops might not be as powerful as assumed. There could be diminishing returns, there could be resource constraints and there could be complexity barriers. So maybe we should just do that kind of recursive self-improving piece first. What do you think about that?
SPEAKER: Neel Nanda
I don't really buy recursive self-improvement. Oh good. It's not an important part of why I'm concerned about this stuff. Um, I so generally I just feel like a lot of the arguments were made before the current paradigm of enormous foundation models When you're investing hundreds of millions of dollars of compute into a thing, it's pretty hard for it to make itself substantially better. Um, and you can do things like design better algorithmic techniques. I think that is probably one that is more likely to be accelerated the better the model gets. And it's not clear to me how much how much juice there is to squeeze out of that. Um, and yeah, um, but generally I just think a lot of this is going to be bottlenecked by hardware and compute and data such that I'm like less concerned about some runaway intelligence explosion and I'm more just concerned about will eventually make things that are dangerous. What do we do then?
SPEAKER: S6
NASA.
SPEAKER: Neel Nanda
And I think this this is like a really good fact about the world. I think a world where you can have intelligence explosions is really scary. And I feel like our current world is a lot less scary than it could have been. If some kid in a basement somewhere just like wrote the code for one day. Yes.
SPEAKER: Tim Scarfe
Yes. Okay. Well, I mean, just just to finish off Katya's final point. So the other point they made was about small differences might lead to over It's a little bit like in squash. I don't know if you've ever played squash, but a tiny difference in ability leads to one player overwhelmingly dominating the other player because you just get these kind of like, you know, it's a game of attrition and you get these tipping points. And she argued that that might not necessarily be the case when comparing systems because of three reasons, different architectures likely to have very different underlying architectures and biological brains which could lead to different scaling properties, performance plateaus. So um, there might be these plateaus beyond which further increases in intelligence don't lead to significant performance improvements. And also this notion of task specific intelligence, something that I strong I believe that all intelligence is specialized as we were speaking about earlier. And so it might be specialized rather than being generally intelligent and small differences thus may not translate into large differences in performance across a wide variety of tasks. Maybe we should just touch on this on this kind of task focus thing. So I think humans are very specialized. We have and we don't realize that we are because the way we conceive of intelligence is anthropomorphic, but actually we don't do four dimensions very well. There's

lots of things that we don't do very well and we're kind of embedded in the cognitive ecology and quite a complex way. So what do you think about that?
SPEAKER: Neel Nanda
Yeah, so I will. Okay. I'll first comment on the general meta dynamic of I think that people get way too caught up on philosophizing and I'm sorry and in particular.
SPEAKER: S7
I care about.
SPEAKER: Neel Nanda
Whether an AI will cause a catastrophic risk. I don't care about whether it fits into whether it's general in the right way, whether it has weaknesses in certain areas, whether it's high on the Chomsky hierarchy or whether it's general intelligence in some specific sense that someone like Gary Marcus would agree with.
SPEAKER: Tim Scarfe
Is is that in any way a contradiction of your mechanistic sensibilities? Because when it comes to neural networks, you want to understand how they work. But when it comes to intelligence, you don't.
SPEAKER: Neel Nanda
Oh, sorry. I want to understand how it works. I want to understand everything. Um, I just don't think it's. I want to disentangle things to be concerned about from theoretical arguments about whether this fits into certain categories for the purposes of deciding whether to be concerned about existential risk. I see all of the theory arguments as like a means to an end of this ultimate empirical question of is this a thing that could realistically happen? And I think that these these theoretical frameworks do matter. Like, I don't know. I think that an image classification model is basically never going to get the point where it's dangerous while a language model that's being left to have some like notion of intentionality potentially will. Um, and. Yeah. I know I can give like random takes, but to me, if you're like I can be tossed specific in the same way that humans are task specific and like, well look, human is task general enough that I think they could be massively dangerous in the right situation with the right advantages like if they wanted to be and were able to run a thousand copies of themselves at a thousand speed or something. I don't know if that's actually a remotely accurate statement about models. Probably they can run many copies but not a thousand x speed or something. But um, yeah, generally. That's the kind of question I care about and I'm concerned many of these definitions lose sight of that. And part of my thing of like, I want to keep a lemon argument as having as few assumptions as possible because the more assumptions you make, the less plausible your case is and the less and like the more room there is for people to like rightfully disagree. I'm like, I want to be careful not to make any of the case. Rest on like strong theoretical frameworks because we don't know what we're doing here enough to have legit theoretical frameworks. And I think that AI is likely to be limited in the same way that humans are at least within the GPT paradigm because if you're training it to predict the next word on the internet and a bunch of other stuff, then it's going to learn a lot from human patterns and human thought and human conventions. But and no.
SPEAKER: Tim Scarfe
In closing, you said that your personal favorite heuristic is the second species argument. Can you can you can you tell us?
SPEAKER: Neel Nanda
Yeah. So, um, I quite like Hinton's recent pithy quotes of there is no example of something being of some entity being controlled by things less smart than it.

And that was terrible.
SPEAKER: Neel Nanda
Uh, sorry.

I really.
SPEAKER: Tim Scarfe
I mean, um, Twitter went wild over that.
SPEAKER: Neel Nanda
I've tried to go.

Wild because.
SPEAKER: Tim Scarfe
I mean, look at and look at a company. The CEO is usually dumber than you have to hire competent people to have a successful company. Or Look at my cat.
SPEAKER: Neel Nanda
Yeah. Okay. Fair. This is a.

Looking.
SPEAKER: Neel Nanda
By the way. Let's just start again. Um. All right. So, yeah, this is often called the gorilla problem, right? Humans are just smarter than gorillas. And basically all ways that matter. Humans are not actively malevolent to gorillas, but ultimately humans are in charge. Gorillas are not. And gorillas exist because of our continued benevolence or ambivalence. And it just seems to me like if you are creating entities that are smarter than you, the default outcome is they end up in control of what's going on in the world and you do not. And I kind of just feel like this should be the null hypothesis. And then there's a bunch of arguments on top of like, Is this a good model? Will Obviously there's lots of disanalogies because we're making them. We ideally have some control over them. We're going to try to shape them to be benevolent towards us. But this just seems like the default thing to be concerned about to me.
SPEAKER: Tim Scarfe
On that point though, we are different from computers. We scuba dive and that's actually quite a profound thing to say. We scuba dive because we are we are integrated into the into the ecosystem, not just physically but cognitively. There's a kind of cognitive ecosystem that we're enmeshed in. We have a huge advantage over computers. Computers can't really do anything in the physical world.
SPEAKER: Neel Nanda
Um, so I agree with this, but I don't know, I feel like the way.
SPEAKER: S8
I don't know.
SPEAKER: Neel Nanda
One evocative example is there was this crime lord El Chapo, who ran his gang from within prison for like many years very successfully. When you have humans in the world, you can get to do things for you. You don't need to be physically embodied to get shit done. And I know just look at Blake Lemoine. There's no shortage of people who will do things if convinced in the right way, even if they know it's an AI.
SPEAKER: Tim Scarfe

And I do agree with you on that. And I think part of the reason why we're going to have the inevitable proliferation of this technology is so many tinkerers will just create many, many different versions of AI and they won't really be thinking about the consequences of their actions. But what's the alternative? Paternalism.
SPEAKER: Neel Nanda
Yeah. So to me the main interesting thing here is large training runs as like the major bottleneck. Very few actors can do them. We're probably going to get beyond the point where people are even putting the things out behind an API open to many people to use, let alone like open sourcing the weights which we've already pretty clearly moved past. And this to me seems like the point of intervention you need if you're going to try to make sure things are safe before you deploy them like track the people who are able to do these runs have standards for what it means to decide a system like this is safe. I'm pretty happy Sam Altman's been pushing that stuff very heavily and if competently done, I think this kind of regulation can be very important. It could be great. Like the Alignment Research Center's been doing great work here and I'm very excited to see what the red teaming large language models thing at Defcon looks like. But and no, maybe me too close. I feel like I've been in the role of why alignment matters. Maybe I can try to break alignment arguments myself for a bit.

Oh, please do.
SPEAKER: Neel Nanda
Yeah. Yeah. So if I condition on Actually the world is kind of fine. Um, probably my biggest guess is that the goal directed notion is just like not remotely a good understanding of how these things work and it's hard to get them to be goal directed and we just mostly coordinate and don't do that. And these systems are mostly just like extremely effective tools. It seems like kind of a plausible world we could end up in. I don't think it's any more likely than yep, they're goal directed and this is terrible. Um, we end up in a world which just has like lots of these systems that don't coordinate with each other, want somewhat different things are like broadly aligned with human interests but like, imperfectly and just none of them ever get a major advantage over the others. And the world kind of continues to be about as the world is with lots of different actors who aren't necessarily aligned with each other, but mostly don't try to take over the world except every so often. Um, or we just alignment isn't that hard. We crack mechanistic interpretability. We look inside the system. We use this to iterate on making our techniques really good. Um, it turns out the doing with like enough adversarial training just kind of works or with AI assistants to help you notice what's going on in the system. And this just gets us aligned human level systems and we can be like, Please go solve the problem. And then they do.
SPEAKER: S9
And I don't know. I think people.
SPEAKER: Neel Nanda
Like Jankowski are very loud about we're almost certainly going to die and that we might, but we also might not. I don't really know. I would love to just become less confused about this and I remain very concerned about this, to be clear. But I'm not like 99% chance we're all going to die.

Yeah, but I mean.
SPEAKER: Tim Scarfe
Anything which is an appreciable percentage may as well be the same thing.
SPEAKER: Neel Nanda

Yeah, pretty much.
SPEAKER: Tim Scarfe
Yeah, it's quite funny. I got a lot of pushback on the Robert Miles show. People said, Oh, I can't believe it. You framed him to be a doomer. And he himself said in the show, I think about five times we're all going to die. And I managed to cut about five. Well, I don't want to exaggerate, but there was at least two posts on Twitter within 15 minutes of that comment where he said, And we're all going to die. So I don't think I don't think I'm being unfair. Well, I didn't actually call him a doomer, but he basically is.
SPEAKER: Neel Nanda
Um, I don't know, man. I hate labels. Like Eliezer is clearly a doomer.
SPEAKER: Tim Scarfe
He's clearly a doomer. Yeah.
SPEAKER: Neel Nanda
Rob is much less doomy than Eliezer. Yeah. Is Rob a dumber? I don't.
SPEAKER: Tim Scarfe0
Know.
SPEAKER: Tim Scarfe
I didn't call him a doomer, but empirically the data says yes.
SPEAKER: Neel Nanda
Um, yeah. I mean, I don't know, man. It sounds like you spend too much time reading YouTube comments.

I do. Too much.
SPEAKER: Neel Nanda
Time. Notoriously the least productive use of time possible apart from hanging out on Twitter reading Flamewars.
SPEAKER: Tim Scarfe
Twitter is the.

Worst.
SPEAKER: Tim Scarfe
Know it's so bad. I mean, we don't we don't need to go there. But we were we were having a brief discussion before, um, before we started hitting record. It's why do you think otherwise intelligent, respectable people behave in that way?
SPEAKER: Neel Nanda
Uh, impulse control. Social validation is just kind of fun. People aren't very self-aware about how they look or like, aren't that reflective. And Twitter incentivizes you to like nuance and to be outraged about other people are I don't know. Um, I am very sad by many Twitter dynamics, including from people who otherwise seem worthy of respect.
SPEAKER: Tim Scarfe1
Yes. Yes. Interesting.
SPEAKER: Tim Scarfe
Look, Neil, this has been an absolute honor.

Thank you so much. Fun.
SPEAKER: Tim Scarfe

Yeah, it's been amazing. It's been a marathon. But thank you so much for joining us today. And I really think we've had a great conversation and I know everyone's going to love it. So thank you so much.
SPEAKER: Neel Nanda
Yeah, I apologize for the times I told you off for philosophizing.
SPEAKER: Tim Scarfe
Oh, no problem. It's it's an honor.

But yeah.
SPEAKER: Tim Scarfe2
All right.
SPEAKER: Neel Nanda
Thanks again for having me on.