Marius' alignment agenda

Meta: this is my current best guess, please feel free to criticize or give feedback. Much of this agenda was inspired by discussions with or work read from other alignment researchers, including Rohin Shah, Evan Hubinger, Chris Olah, JS Denain, Stephen Casper, Dan Hendrycks, Tom Lieberum, Daniel Ziegler, Neel Nanda, Adam Scherlis, Buck Shlegeris, Beth Barnes, Ajeya Cotra, Danny Hernandez, Richard Ngo, Mark Xu, Paul Christiano, John Wentworth, Quintin Pope, Jacob Steinhardt, and more.

A lot of the following was inspired by a chat with Rohin Shah about his agenda (at least the science of DL part). It is very likely that the details diverge from what Rohin intended, so you should not infer the quality of his agenda from this doc. Also, the more I thought about the agenda, the more it looks like Chris Olah's :)

Update: I now have a more specific agenda that I'm pursuing at Apollo. You should not assume that I still stand behind all claims in this post.

Note to the reader

This agenda is by no means static or finished. Feedback and pointers to further resources are very appreciated. **I'm certainly missing a lot of relevant related resources and am not the first person to have these ideas (in many ways this agenda is inspired by or related to work done at DeepMind, Redwood Research, ARC, Anthropic[1] and others)**. If you feel like my ideas are wrong or could be better, please let me know - **Feedback and scrutiny are welcome**.

I'm also **interested in collaborations** with other researchers or am willing to supervise work on this agenda (I've received feedback that I'm a good manager/mentor from multiple independent sources). In case you like the agenda and think I should work for you, feel free to reach out.

Obviously, I don't own this research! You don't have to ask for permission to work on it and having multiple people try the same thing independently seems helpful (collaborations and coordination is probably net beneficial though).
If you're already working on something that relates to the document feel free to reach out for coordination even if you don't want to collaborate. I'm aware that this agenda is way too much for one person and I'll likely only be able to focus on specific parts of it.

[1] Anthropic's and Redwood's agenda is probably the closest to what I have in mind. I expect there to be some differences but I don't expect them to be huge. The main difference is that I'd be more keen on applying the interpretability tools that currently exist to try and understand DL with a bit less focus on developing new methods. However, this might change, e.g. if current methods turn out to be insufficient. The two pieces of research that are currently closest to what I imagine the work to look like are Neel Nanda's and Tom Lieberum's work on grokking and Anthropic's work on polysemanticity.

Motivation

There are many possible ways in which advanced AI could go wrong.

- Advanced AI without countermeasures will likely lead to takeover
- Advanced agents are likely power-seeking
- Advanced agents might get a crucial fact about the world wrong and thus accidentally cause extinction
- Aligned powerful systems might be extorted if their utility function is easily inferable and not robust (see e.g. CLR's agenda).
- Advanced AI systems might be deceptively aligned, e.g. they appear aligned but are actually not (see e.g. Evan Hubinger's post, Evan's new post).
- Unknown unknowns: there are likely failure modes that we fail to anticipate.
- Many more (see e.g. Neel Nanda's overview of the alignment landscape, Evan Hubinger's overview of 11 proposals to address some of these risks, Paul Christiano's overview, Rohin Shah's overview, Richard Ngo's AGI safety from first principles)

My thoughts on these include:

1. **All of these problems seem significant and important.** I think "getting a crucial fact wrong" is the least problematic failure mode because gains in capabilities might solve this problem already, e.g. more powerful models are likely to be more knowledgable or better calibrated on their own uncertainty.
2. **Deceptive misalignment seems the most relevant to me**: I have detailed my reasoning in this LW post. In short, I think most of the big risks come from situations where the AI system actively tries to hide its goals and deceive us while actually pursuing a different goal in the background.
3. **All or most of these problems would be easier to solve if we understood the underlying AI systems and the process that creates them better**. This is also true for lots of other problems unrelated to AGI safety such as fairness or biases in ML. It's just easier to point out and fix problems if we understood the model's beliefs and the process by which they are created.
4. **Understanding AI systems better seems to be helpful even if none of the above problems turn out to be relevant** (as was already pointed out by Neel Nanda for interpretability)
5. **Science of DL is a broad approach.** Like I said before, understanding the system better nearly always helps to align it and this holds pretty independently of the exact system you're trying to align. Furthermore, there are few diminishing returns on how many interpretability researchers are useful, e.g. even if there were tens of thousands of people who have a deep understanding of understanding DL systems and NNs that would still seem net positive for alignment.

In short, you could also say that I now think that many people before me were right (see here, here, here, here, here, here, here or here), i.e. that deceptive misalignment is where a lot of the risk comes from and interpretability or "understanding the AI system more broadly" is the most promising answer we currently have.

Overview - Science of Deep Learning

By *Science of DL,* I roughly mean “understanding DL systems and how they learn concepts” better. The main goal is to propose a precise and testable hypothesis related to a phenomenon in DL and then test and refine it until we are highly confident in its truth or falsehood. This hypothesis could be about how NNs behave on the neuron level, the circuit level, during training, during fine-tuning, etc. This research will almost surely at some point include mechanistic interpretability but it is not limited to it.
The refined statement after investigation can but doesn’t have to be of mathematical form as long as it is unambiguous and can be tested, i.e. two people could agree on an experiment that would provide evidence for or against the statement and then run it.[2]

How this would look like in practice

The details would obviously differ from project to project but on a high level I imagine it to look roughly like this

1. **Pick an interesting concept found in deep learning**, e.g. grokking, the lottery ticket hypothesis, adversarial examples or the emergence of 2-digit addition in LLMs. Optimally, the concept is safety-related but especially in the beginning, just increasing general understanding seems more important than the exact choice of topic.
2. **Try to understand high-level features of the phenomenon**, e.g. under which conditions this concept arises, which NNs show it and which ones don’t, in which parts of the networks it arises, when during training it arises, etc. This likely includes retraining the network under different conditions with different hyperparameters, number of parameters, etc. and monitoring meaningful high-level statistics related to the concept, e.g. monitor the validation loss to see when the model starts to grok.
3. **Zoom in:** try to understand what happens on a low level, e.g. use mechanistic interpretability tools to investigate the neurons/activations or use other techniques to form a hypothesis of how this specific part of the network works. In the optimal case, we would be able to describe the behavior very precisely, e.g. “this is a car circuit” or “this circuit models addition”.
4. **Form a testable hypothesis:** Once we feel like we understand what’s going on for this particular part of the network, we form a testable hypothesis. This could be a hypothesis about how networks learn something, e.g. “when X happens during training, we will see more of this phenomenon”, or about concepts that relate to the part of the network, e.g. “this is a circuit related to animals, let’s see if it lights up when you talk about a ‘cuckoo clock’ (which is not an animal; just a specific kind of clock)”.
5. **Test and refine the hypothesis:** Test the hypothesis and attack it from multiple angles. Try to find corner cases and actively play an adversary role, e.g. by suggesting alternative explanations for the phenomenon. Use the process to refine our

[2]these experiments can also be thought experiments but in the context of DL empirical experiments are often possible

understanding and propose a new testable hypothesis. Repeat until we're sufficiently confident.

6. **Generalize:** Make a speculative claim that might or might not be implied by the more narrow hypothesis we are relatively confident in. Theorycraft why this speculative claim relates to the narrow hypothesis. Once we have a plausible theory for why the speculative claim could relate to the previous hypothesis, we translate it into a new testable hypothesis (the theorycrafting is necessary so that we are forced to build mechanistic mental models of how DL works).
7. **Iterate:** Repeat the above steps as long as it makes sense, e.g. for new concepts and settings.[3]
8. **Get fast and automate:** I think the goal should be to "understand" important components of a neural network very fast, e.g. it takes one human (with automated tools) less than 24 hours. For this to be successful, we need to train the skill of understanding a neural network and we need automatic (narrow/harmless) tools to assist us.

Goals

The goal of this research is to understand DL systems as well as possible. This means there is not one clear goal by that we could judge our performance. However, I think there are some ways to test whether we actually increased our understanding of different parts of the system. These include

1. Can we predict with high accuracy whether a network will learn or not learn a specific property before we train it, e.g. from size, hyperparameters, data and compute alone?
2. Can we predict with high accuracy whether a phenomenon has already been learned before we test it on a benchmark, e.g. can we tell whether it learned 2-digit addition after looking at a set of circuits?
3. Can we predict with high accuracy which part of the network is "responsible" for a task with a very limited budget of forward passes, e.g. if we're only allowed ten different prompts? Bonus: can we explain the respective behavior?
4. Can we attribute a specific input-output behavior and explain it on a mechanistic level within a certain budget of time, e.g. can we find the "car circuit" in an LLM within 60 minutes using whatever method we want?

Caveat

Understanding more parts of the DL pipeline can always also lead to an increase in dangerous capabilities. Essentially, whenever we understand technology better, we can use that knowledge to make it more efficient or powerful.

However,

[3] "Makes sense" is mostly defined by how relevant it is to understand the network relative to doing other research, e.g. at some point we might hit diminishing returns.

- I think that understanding the system better tends to favor alignment vs. capabilities, i.e. for alignment understanding seems more necessary than for capabilities (see e.g. my post on the defender's advantage of interpretability),
- since people deploy ML systems in the real world at large scale, I don't really see a way around "understanding the system better" and
- one can always choose not to publish the results or only share them among a trusted group of researchers. I expect at least some of this work to be private by default.

Overview - Auditing AIs

The gains in understanding from Science of DL need to be translated to real-world applications of AIs to lead to gains in safety. I think the obvious and also most pressing application is AI auditing.

Over the next years, companies will deploy more and more powerful AI models in the real world. So far, every company does its own auditing to different degrees. Some companies take it very seriously and others don't. Therefore, I think there is an important niche to be filled in developing reasonable auditing protocols.

There are many challenges in developing good auditing protocols since we don't have clear visions of what we actually want to audit for and even if we knew we often lack the tools to do so. There is a second challenge that whatever protocol someone comes up with has to be realistic enough that people buy into it without being useless. Finding this trade-off seems hard.

On the other hand, if we were able to develop a protocol that is reasonable and gets buy-in we might learn many things about auditing more powerful AIs in the future and we might have already set up the network necessary to audit future models.

Comment: *I have thought less about the auditing section than the previous one and it is more about asking lots of questions than providing approaches to answer them. If you have answers to them or links to resources (even very basic ones), I'd be delighted to read them.*

Table of contents

- How exactly do **transformers learn a concept** during training? Investigate the inductive biases of transformers in more detail. Are there theoretical or practical limits to the attention mechanism? Can these be resolved by stacking more transformer blocks on top of each other? (skip to section)
- A possible first project - **Throwing the kitchen sink of interpretability methods on an MNIST classifier** (skip to section)

Auditing AIs:

- Developing a realistic and helpful AI auditing protocol (skip to section)
- Benchmarking alignment in LLMs (skip to section)

Misc.:

- Can any of the above yield a practical implementation of **ELK**? (skip to section)
- (TBD) Investigating causal understanding in LLMs (skip to section)
- (TBD) General DL theory (skip to section)

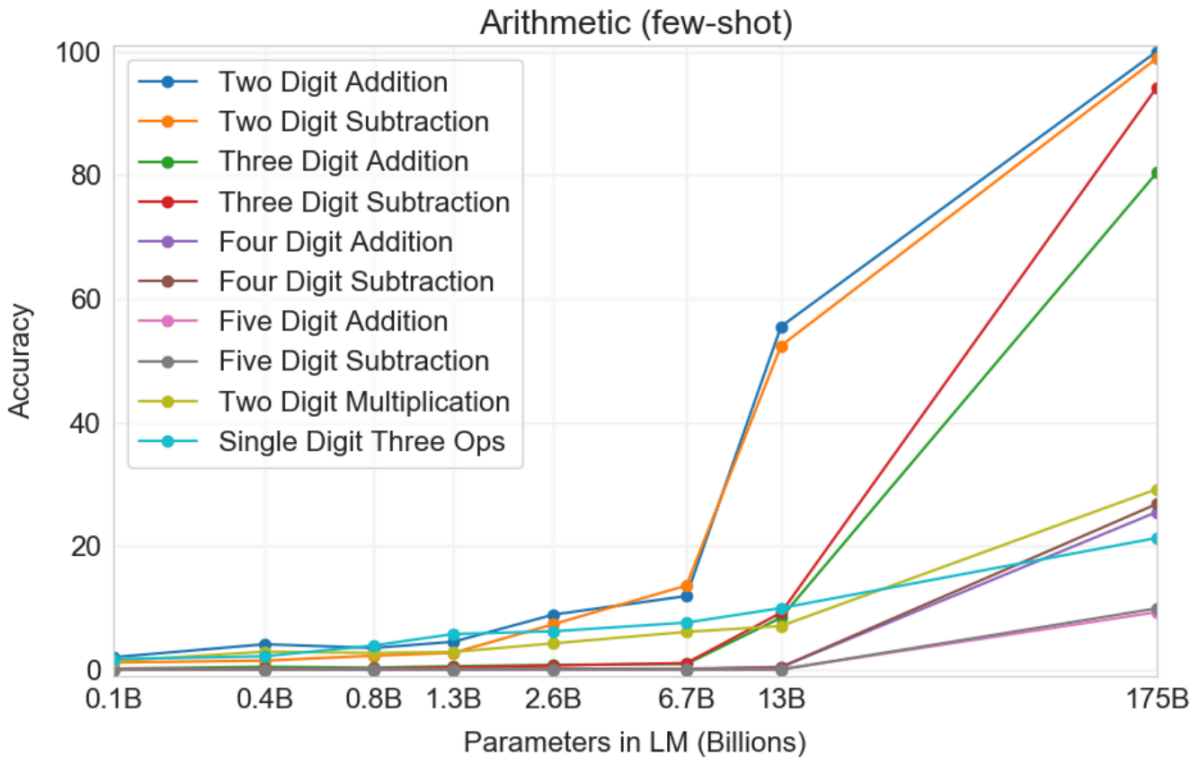
World-view investigations type work:

- (TBD) Measuring algorithmic progress (skip to section)
- (TBD) Measuring training-inference trade-offs in DL (skip to section)

Details - Science of Deep Learning

Explaining emerging phenomena in transformers

There are a number of emerging phenomena in transformers (and in DL in general), i.e. when trained with X parameters an NN does not have a specific capability such as two-digit addition but when trained with 10*X parameters it suddenly does. In the case of GPT-3, for example, larger models are much better at N-digit addition and subtraction tasks (see figure below).



Given that many of the phenomena AI alignment people are worried about (e.g. modeling humans well, understanding value systems, being deceptive, etc.) might also "just emerge" with scale, it seems pretty important to understand how exactly they emerge. Potential research questions include

- **What kind of scaling leads to any given phenomenon**, e.g. is it more parameters, more compute, more/better data, a combination of those or something else? In case it comes from more parameters, do we know which kind of parameter scaling leads to this capability, e.g. does it come from more attention heads, wider layers, deeper transformers or something else?
- **How "step-function-like" is this phenomenon in the first place?**[4] There is a lot of space between X and 10*X parameters, so the emergence of the phenomenon could be explained by a linear or exponential function as well as a step-function. All possible results have interesting implications for follow-up experiments, e.g. in case it's step-function-like, it would be interesting to investigate models right before and after the step.
- Can we understand which **combination of attention heads** is responsible for the specific phenomenon in the transformer of size X and size 10*X? If yes, what is the

[4] I think the BIG-bench paper claims that most emergent phenomena are not really emergent if you look close enough at the right metric (see Figure 8.) https://arxiv.org/pdf/2206.04615.pdf But I'm not sure I understand their claim correctly. Just skimmed it.

difference between the respective attention heads? Are there specific patterns, that can explain the improved abilities?
- Can we **build a low-level mechanistic model** of this phenomenon that we can translate into human concepts? Do we understand the network well enough that we could modify it and accurately predict its change in performance?
- Can we point to the **part of the network** that is primarily responsible for the emergent phenomenon? Can we derive any general rules from this, e.g. that deeper layers correspond to more abstract abilities?

Related work:
- Emergent Abilities of Large Language Models [arxiv]
- Formal algorithms for transformers [arxiv]
- Scaling Laws for Autoregressive Generative Modeling [arxiv]
- GPT-3 paper [arxiv]
- GPT-2 paper [arxiv]
- PaLM paper [arxiv][blogpost]
- Chinchilla paper [arxiv][blogpost]
- Memorization Without Overfitting: Analyzing the Training Dynamics of Large Language Models [arxiv]
- We might be able to see the sharp left turn coming [AF]
- Predicting Broken Scaling Laws [arxiv]

Explaining what happens during RLHF or adversarial training

Reinforcement learning from human feedback (instruct GPT[5]) and Adversarial training for high-stakes reliability are two black-box approaches to alignment. In both cases, we train the model in specific ways or on specific datasets to show good behavior. In both cases, I was surprised by how little data was necessary for relatively big gains in performance. RLHF seems to train on 50k-100k prompts (see appendix A.3 in RLHF paper[6]) and yielded a more than 10pp improvement in comparison[7] to a baseline GPT (even the instruct-GPT-1.3B outperforms the 175B baseline).
Redwood's adversarial robustness approach uses ~20k training prompts[8] and they claim “With our chosen thresholds, filtering with our baseline classifier decreases the rate of unsafe completions from about 2.4% to 0.003% on in-distribution data, which is near the limit of our ability to measure.” This seems like quite the improvement for comparatively little fine-tuning.

---

[5] Depending on who you ask, this is not alignment. I probably belong to the camp that considers it capabilities.
[6] Jan Leike stated that GPT-3 was fine-tuned with RLHF on less than 2% of the compute budget that was used for pre-training.
[7] This sounds impressive to me, but I'm not sure how good that actually is. In some sense, it's just a bit better than guessing.
[8] I think ~200k prompts in general but only ~20k of them contain violence. I find the phrasing a bit unclear.

Since both of these approaches will likely be used in practice, it seems important that we understand them much better than we currently do. Possible research questions include

- **Investigate the same things that are already explained in the section on emergent phenomena.** For example, pre- and post-fine-tuning attention patterns, how step-function-like the capability really is, etc.
- Investigate which **parts of the network** are most affected by the respective approach. For example, if later layers are more affected, this might indicate something about which kind of knowledge is stored in which parts of the network. For example, more high-level concepts might be stored in later layers similar to CNNs. This could also give us relevant insights into which parts are responsible for vague and abstract concepts such as violence and thus make interpretability techniques easier to use.
- Look at the exact **scaling of the approach**, i.e. how much does the relevant metric change if we use 10, 100, 1000, 10k, etc. prompts? Can we infer a scaling law for both approaches somehow? What does this tell us about the limit of both approaches?

Related work:

- Instruct GPT [blogpost]
- Adversarial training for high-stakes reliability [arxiv]

Understanding Grokking

Grokking describes the phenomenon that NNs first overfit on the training data and then generalize after many epochs (e.g. 100k or more). So far this phenomenon has mostly been seen on small transformers on algorithmic datasets by using large learning rates and large weight decay. This project would mainly revolve around really understanding this phenomenon in more detail, e.g. answering lots of basic questions. *There are some other people working on this problem that might be interested in collaborations or help out. These include a small team at OpenAI, a small team at DeepMind, Tom Lieberum, Neel Nanda, Buck/RR and possibly a team at Anthropic.*[9]

- Update: the omnigrok paper has answered some of my questions but still not the most important one, i.e. what changes during the training on a mechanistic level that makes us understand how the model generalizes.
- What seems important is how these more general circuits form during training and in general. So the important bit seems to be interpreting not just the end product but the entire training run and see if we can learn anything (see Tom's comment for more).
- How does the attention pattern of the transformer change in the grokking epochs, can we explain what that means? Look deeper into transformations of the attention pattern (see Neel Nanda's and Tom's work)
- How do the activations of transformers change during the grokking epochs? Do the activations correlate with grokking?
- How do the gradients (aggregates and individual gradients) change over time? Do the gradients correlate with grokking?

[9] Mostly speculation. No hard evidence

- Can we find any property that would be predictive of grokking that is not the validation error, e.g. gradients, weight norms, activations or a combination of them?
- Can we reproduce grokking with slightly different hyperparameters, e.g. lower learning rate or lower weight decay? If not, what property of these hyperparameters leads to grokking?
- How does the norm of the difference between weights change between epochs? Do large changes correlate with large grokking updates?

Related work:

- Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets [arxiv]
- Neel Nanda's work on Grokking [AF]
- Hypothesis: gradient descent prefers general circuits [AF]
- Omnigrok: Grokking Beyond Algorithmic Data [arxiv]

Interpretability

Interpretability seems like a very helpful component of many different alignment approaches. However, I think there are a couple of shortcomings of current interpretability techniques. Firstly, the concept of interpretability is a bit vague/subjective and thus hard to measure with more objective metrics. Therefore, concretizing and measuring specific aspects of interpretability seems like an important first step. Secondly, most current interpretability approaches focus on explaining smaller networks in great detail. If we ever want to make interpretability useful on large NNs, we have to come up with ways to scale it.

I think the interpretability overview is really helpful and makes many suggestions in the final section. These claims include ideas on measurability and scalability but go even beyond that. Therefore, if one would start to work on these ideas, reading this section is probably helpful.

Related work:

- A Mathematical Framework for Transformer Circuits [website][youtube]
- Toward Transparent AI: A Survey on Interpreting the Inner Structures of Deep Neural Networks [arxiv]

Measuring/benchmarking interpretability

Stephen Casper has thought about a specific part of this question in more detail. I think his ideas and approach sound promising.

Stephen Casper's take: "From first principles -- what we want from interpretability tools are insights that are both *valid* and *useful* for working with models.
Many measures of success that past works have used for interpretability tools are very non-rigorous and/or fail to meet this standard. For example, "the visualization of the neuron looks like a dog," "The neuron responds the most to dogs in this dataset,", "We had people use this tool, and they think that this neuron is a dog neuron," etc. The first issue here is that I think

methods like this are hypothesis generators that are often treated as conclusion generators. The second is that -- even if people use an interpretability tool to make and validate a testable prediction -- I fail to see much meaning in an interpretation like this anyway unless it's connected to something I might care to do with a model.
What are good examples of success measures? I think that some include using an interpretability tool to guide the design of a novel adversary, manually finetuning a model to induce a predicted change in behavior, reverse-engineering a model, or similar. I think that interpretability benchmarks should measure this kind of usefulness.
My current work involves introducing a variety of trojans into image classifiers and evaluating applicable interpretability tools (most of which involve constructing adversaries) by their ability to help m-turkers who do not know about these trojans to rediscover them. This gets at measuring both of my desiderata for good interpretability tools -- validity and usefulness. Very similar work could also be done for LLMs"

More concrete questions include (I don't claim that these are new; some of them are probably already done in practice)

- **Find more precise definitions of interpretability:** One of the reasons why interpretability is hard to measure is because the concept encompasses many different components and is very subjective, i.e. it is often defined in terms of what humans would think about a specific NN. Therefore, it is hard to find a ground truth for interpretability. One possible conclusion is that interpretability is an ill-defined concept. Thus, thinking about specific subcomponents of interpretability or just making it more precise might already naturally lead to more measurable concepts. For example, do two people have to have the same interpretation of a network for it to be a valid interpretation or does at least one of them have to be wrong? Is compression a necessary component of interpretability?
- **Use interpretability-adversarial duality** to measure specific aspects of interpretability. This would essentially boil down to taking Stephen Casper's ideas and translating them from CNNs to LLMs. Since he is still at an early stage it probably makes sense to wait with this project until he has made more progress and then translate the findings.
- **Test hypothesis on held-out test data.** Just as in conventional NN training, we could interpret the meaning of different neurons/activations on our training data and then use a held-out test dataset to validate/falsify our hypothesis. An informal way of testing this would be to test if the neuron that was identified as the dog-neuron during training still represents a dog-neuron during testing. A more formal way would be to measure the size of the activation on data points that clearly encompass the respective features, e.g. whether the dog neuron is activated as strongly, as it was on the training data (to remove some degree of subjectivity). There are many other ways to test interpretabiltity that I'm excited about. I'm mostly saying that we should start testing it more rigorously in the first place.
- **Measure agreement between different interpreters:** Let multiple different people interpret the same model and measure correlation/other similarity measures to compare their annotations. This could also give insight into how clearly a node represents a specific concept, e.g. when all people agree, the concept is represented more clearly

than if only 50% of annotators agree. This “clarity” of interpretability is another possible metric to report (related to Anthropic’s SoLU paper)

- **Train the same architecture on the same data with slightly different hyperparameters** and annotate every different resulting network. One could change a combination of random seed, learning rate, momentum and weight decay and then measure the similarity/difference between the resulting concepts of the networks. This could give insight into which concepts are “necessary/universal” to represent the training dataset and which ones are not (as hypothesised by Chris Olah).
- **What is the relationship between interpretability and counterfactuals?** There is an argument that interpretability comes, to a large extent, from counterfactual reasoning and I think this is plausibly a core component of interpretability. Buck Shlegeris and Quintin Pope have thought about this.
- **Play the auditing game:** As described in Automating Auditing: An ambitious concrete technical research proposal, we could play the auditing game to measure how well a specific interpretability technique works. In short, an attacker modifies/fine-tunes a NN such that it produces some unintended behavior. Then, the auditor has to explain which behavior was modified (and possibly point to the responsible weights in the NN). Optimally, we would want the auditor to be automated. If an auditor can consistently win the game, we can see this as evidence that they have effective interpretability techniques.

Related work:

- Softmax Linear Units [blogpost]
- Two papers by JS Denain:
    - Auditing Visualizations: Transparency Methods Struggle to Detect Anomalous Behavior [arxiv]
    - Grounding Representation Similarity with Statistical Testing [arxiv]
- Suggestions by Stephen Casper:
    - Debugging Tests for Model Explanations [arxiv]
    - "Will You Find These Shortcuts?" A Protocol for Evaluating the Faithfulness of Input Salience Methods for Text Classification [arxiv]
- Text Counterfactuals via Latent Optimization and Shapley-Guided Search [arxiv]

Scaling interpretability

If we ever want interpretability to be useful in crucial real-world tasks, we have to scale it to larger models. The goal I have in mind here is something like: *being able to interpret all ‘relevant’ parts of a large NN, e.g. GPT-3, in 24 hours or less.* This is clearly ambitious given the current capabilities of interpretability but I don’t see a way around it if we want interpretability to make a difference for alignment.

Possible ideas include:

- **Automate interpretability:** One possible way to scale interpretability is to understand it very well on a small scale, use the small-scale annotations as labels and then use an

additional NN to automate it. This interpreter network can then be used to automatically annotate large-scale NNs much faster than any human could. I'm not sure how well this approach works since large NNs might have either different or much more fine-grained concepts than small NNs.

- **Use automatic caption generation systems for interpretability:** There are already a number of decently working caption generation systems for real-world images and there are summary systems for text. I could imagine that large LLMs fine-tuned on summary could already be pretty good at generating concepts that describe particular neurons. In computer vision, the work of Zeynep Akata might yield some valuable insights.
- **Use high-level statistics to select 'relevant' parts of NNs:** Often we don't need to understand the entire network but only some 'relevant' parts. This could mean, that we want to interpret the parts of the NN that contribute to a specific behavior or output, e.g. the cumulant-related ideas that Redwood is currently working on (public soon). This could also mean, that we are only interested in the parts of a NN that are relevant for alignment, i.e. most of the neurons (like the dog-neuron) have a low likelihood of being relevant to alignment. Thus, having a mechanism to identify and zoom in on only the alignment-related concepts would cut down the necessary work by a lot. I'm not sure there is a clear distinction between alignment-related and non-alignment-related concepts or that we can realistically implement it[10].

Related work:

- For computer vision, Zeynep Akata's work might be inspiring [google scholar]
- Redwood's work on cumulants [not yet public]
- Redwood's work on interpreting one circuit of GPT2-small [openreview]
- Moving the Eiffel Tower to ROME: Tracing and Editing Facts in GPT [arxiv]
- Mass-Editing Memory in a Transformer (follow-up to ROME paper) [arxiv]
- SVD for transformers [AF]

How do transformers learn concepts during training? Inductive biases of attention

The previous questions are sub-questions of this one. In this section, I'm interested in a more high-level understanding of transformers and attention. On one hand, I'd like to have a better understanding of how concepts are learned during training. This question is strongly related to the previous two questions and the experimental setups are roughly similar. On the other hand, I'd like to understand what inductive biases transformers and attention mechanisms have in general. **Understanding the inductive biases of transformers is important because of deceptive misalignment** (as argued e.g. by Evan Hubinger).
Possible research questions include

[10] If I wanted to investigate whether a human is aligned with my values, I can often produce a very good guess without understanding the vast majority of his values or concepts. This gives me some confidence, that we can find methods to investigate important properties that don't require a perfect understanding of all neurons and circuits.
The question is if "a very good guess" is enough

- What are the theoretical limits of a single attention block? Are there patterns it can clearly not learn similar to how e.g. single layer linear NNs can't learn XOR.
- Does the combination of attention layer + linear layer + non-linearity have any fundamental limits to what it can learn?
- Does the combination of multiple attention blocks have theoretical limits?
- What are the practical limits of multiple stacked attention blocks, e.g. can all possible patterns be learned from generic text data? Which data would a transformer need to see to learn a specific pattern?
- Can we verify all of the above empirically?
- Which kind of patterns do transformers learn empirically? This is very related to the work of Anthropic and their findings of e.g. induction heads, etc.
- Which kinds of patterns are being learned in which layers? Can we find a plausible hypothesis for why that is, e.g. rising complexity with higher depth (similar to CNNs)?
- What kind of inductive biases do different parts of the training pipeline contribute to the process, e.g. data, architecture, reward, or training-procedure? For example, I expect that some of the things we currently don't understand about LLMs will look quite straightforward once we investigate the trillions of tokens they are trained on in more detail.

Related work:

- Do Language Models Have Beliefs? Methods for Detecting, Updating, and Visualizing Model Beliefs [arxiv]
- In context learning and induction heads [blogpost]
- Is depth useful for self-attention? [blogpost]
- Limits to Depth-Efficiencies of Self-Attention [arxiv]
- Theoretical Limitations of Self-Attention in Neural Sequence Models [arxiv]
- Attention is Turing Complete [JMLR]
- Attention is all you need (already discusses some limitations AFAIK) [arxiv]
- Overcoming a Theoretical Limitation of Self-Attention [arxiv]
- Transformers are sample efficient world models [arxiv]

A possible first project - replicating interpretability results

To execute this agenda, it is necessary to understand many interpretability tools well, to develop intuitions for how useful they are and develop fast pipelines to evaluate the interpretability results in quantitative, non-subjective ways. To achieve these goals, I intend to start with a simple model and try to interpret is as well as possible. There is some work that training on adversarial examples and sparsity improve interpretabilty. I intend to replicate some of these results in simple settings. Importantly, I don't expect any relevant results from this work. It's mostly a way to get better with interpretability tools on a problem that allows for very fast iterations.

Concretely, this would look like

1. Using as many interpretability tools as possible (e.g. taking them from this overview) and apply them to a simple MLP or CNN trained on MNIST. In other words, I want to "throw

the entire kitchen sink of interpretability tools" on a simple model. It is possible that models trained on MNIST are not very interpretable and I might have to switch to other datasets.
2. Benchmark the different tools using non-subjective, quantitative measures (if the currently available ones are insufficient, I'll have to come up with some basic new ones. But just making quantifiable predictions about activations and measuring them on test data sounds like a plausible start; suggestions welcome).
3. Develop an intuition for the different interpretability tools. I expect some of them to be very useful and most of them to produce only marginal increases in interpretability (as is true for most approaches in ML). I just don't know which one is which yet.
4. "Fix" some of these networks, i.e. understand them well enough to increase their accuracy. Concretely, I'd be interested if I'd be able to get to a point where I can "read" the activations on the training set (especially where the network misclassifies a number) well enough to change the respective weights to increase training accuracy. Not sure if this works but it's a good benchmark to aim at.
5. Replicate some of the interpretability work on transformers, e.g. Neel and Toms analysis on grokking, Eric's work on grokking or Anthropics work on transformer interpretability. For much of this work, the code is public, so I should be able to replicate their findings.
6. I have already started with some of this work and so far I'm moving faster than expected and interpretability algorithms are easier to implement than expected. So, at the moment I'm cautiously optimistic that this project could actually lead somewhere.

Related work:

- Mathematical Circuits in Neural Networks [LW]
- Work on grokking (see section)
- Anthropics interpretability work for transformers

Auditing AIs

The core reason why I want to do Science of DL is that I want to understand the models better. The core reason why I want to understand the models better is that I want to make sure they are safe. The most obvious way in which we can leverage this understanding is by auditing AIs at every step of the pipeline and from many different angles.

Comment: *I have thought much less about this section than the previous one and it is more about asking lots of questions than providing approaches to answer them. If you have answers to them or links to resources (even very basic ones), I'd be delighted to read them.*

Developing a realistic and helpful auditing protocol

It seems obvious that future models have to be audited before release. The question is just what a good auditing protocol should look like and which kind of institutions are responsible for applying it. Some of my questions include:

- Which kind of things do we want to look out for?
    - Social desiderata, e.g. model doesn't produce violence, isn't racist, sexist, etc.?
    - Is aligned to "good values" (whatever these may be), e.g. doesn't assist in making bombs or is not deceptive?
    - Is the model helpful to humans, e.g. as rated by human m-turk workers?
    - Is the model intent aligned, e.g. does it exactly what we want it to do?
    - Which of these do we even want to measure? Just all of them? Which are more important than others?
- How do we look out for these things?
    - Just run the model on a ton of high-quality benchmarks?
    - Should mechanistic interpretability play a role in auditing? In the long run, it probably should. In the near term, it might not be feasible.
- Who does the auditing?
    - Would someone provide a protocol that individual actors can apply themselves?
    - Would there be an organization that does the auditing such that it is harder to game it?
    - I see the advantages and disadvantages of both possible options. However, a well-run and well-trusted auditing organization might have very positive effects in the long run, so if there are good people for that, this seems like the better option.
- How would we create norms in industry and academia such that thorough audits become the default?
    - Some organizations already put structures and protocols in place to audit their models. The Gopher model, for example, seems to have been trained roughly one year before it was released, implying that DM took safety concerns very seriously. Stable diffusion, on the other hand, likely had zero auditing and was released with model weights and a detailed description of how to use it without any concern for misuse or accidents.
- I could imagine working for/leading/founding such an organization in the future but I'm currently clearly not knowledgeable or experienced enough and don't have the necessary social capital to do so.
- How do other communities handle auditing? The aviation industry is often used as a poster child for many aspects of security and public relations. However, I don't have a detailed overview of how the aviation industry or others handle auditing but some best practices are surely translatable.

Main goals

I think the main goal is to produce a reasonable protocol that people actually use. At the current stage, it might even be more important to get people on board with the idea that auditing is useful and establishing auditing as the norm than whether the protocol is perfect.

The other goal is obviously to develop a reasonable auditing protocol.

Main difficulty: I think the main difficulty for Auditing is the following. In the beginning, an auditing protocol has to be easily accessible to get buy-in from the community. This means that

it will primarily benchmark the behavior of the model instead of opening the black box because interpreting a network is time-intensive and harder than running a benchmark. With a well-designed benchmark, we could understand the input-output behavior of the model well. However, IMO the largest risks from misaligned AI likely come from deception. Thus, such a protocol will have to make the transition from primarily measuring input-output behavior to measuring internals as well. During this transition, the protocol or relevant organization would still have to keep the buy-in of the community even though the new protocol is more costly. Making this transition seems hard but also necessary and I expect it to be one of the hard tasks for someone developing an auditing protocol or leading an auditing org.

What do we want to measure?

Roughly speaking, it seems like there are two paths to auditing AIs.

1. **Black-box auditing:** this mostly consists of monitoring the behavior of the AI, e.g. the outputs it produces on a given benchmark. In practice, this could look like a BigBench but for alignment-related tasks. I have talked to some people in the alignment community about this idea (see next section) and there seem to be lots of disagreements about which kind of tasks are actually useful to measure.
   I currently think we should just measure lots of things and see how we can improve our benchmarks and make sure they actually measure a relevant concept, i.e. I'm for a "move fast and test things" approach. Once the benchmarks are good enough that they are internally useful for alignment orgs, one can still think about how to make them useful and accessible to other organizations.
2. **Internal auditing:** Just measuring the input-output behavior of NNs seems insufficient for many realistic risks from AI safety. Therefore, it makes sense to open up the black box and use interpretability tools to understand the internal beliefs, algorithms, etc. of the model. I think current interpretability tools are still too limited to actually have a reasonable protocol right now, so the protocol has to wait until we have better tools for interpretability. However, I think there are a couple of interesting ideas that could already be done. For example, one could seek to define more objective metrics for interpretability and then kick off some Kaggle competitions along the lines of "whoever interprets the network the best wins" or "whoever interprets the network the fastest wins" or "whoever explains most of the network given a specific budget of forward passes wins". These competitions could spur more interest in making interpretability fast and actionable.

How do we get people to use it?

Make it good, make it reasonable, make it easy to use. Not sure in which order.

Related work:

- I think many of the big labs, e.g. OpenAI, DeepMind and Anthropic, have thought about auditing a lot but I don't know how public their thoughts are.

Benchmarking alignment in LLMs

One part of science is to measure things more accurately. In this particular case, we might want to measure the alignment-related capabilities of LLMs. I wrote a google doc with different suggestions for possible ways to measure alignment in LLMs and multiple people in the alignment community have given their takes on it (if you want to have access to the detailed doc, PM me).

I'm currently much less excited about the project than I originally was because

1. There seems to be little agreement within the alignment community on what a good alignment benchmark for LLMs looks like and most suggestions are seen as net negative/accelerating/deceptive by at least one person (and I find the reasons somewhat convincing but with high uncertainty).
    a. Concretely, the biggest disagreements seem to be between value alignment and intent alignment. People who think that value alignment is the thing we should measure (e.g. which kind of moral framework an LLM is using) often stated that intent alignment (or at least many forms of intent alignment) would effectively be capabilities research and thus shouldn't be done. People who think that we should measure intent alignment (e.g. whether the LLM does what the user intended it to do rather than something slightly different) often think that value alignment benchmarks will lead to more zero-sum mentalities where one party wants to align their AI before everyone else does to get it to have their values.
    b. I'm currently not sure which argument I find more persuasive. I think we won't get around measuring both aspects (values and intentions) to some degree. The main takeaways for me are
        i. that we should gain clarity about what we actually want to measure and
        ii. that not all benchmarks should be public (at least upon release).
        iii. We will have to start building benchmarks and measuring results even if not everyone agrees on the path. Getting more info on the behaviour of current models is urgently necessary. In case the benchmark is controversial, just don't publish it--but definitely build it (after getting feedback, etc.; don't build it alone in your basement).
2. Creating and maintaining a benchmark is very hard especially if it's not in your primary field of research. Thus, I feel like I'm probably the wrong person to do this and other people are a better fit. I might change my mind on this in the future and I think it wouldn't take me very long to become a good fit if I wanted to (maybe a month or so with tips from people who have more experience).
    a. Dan Hendrycks has some very helpful tips on building benchmarks

Related work:

- Dan Hendrycks has built many benchmarks including some for alignment [Google scholar]
- Owain Evans has built benchmarks for alignment, most prominently truthfulQA [arxiv]
- Beth Barnes is building benchmarks for alignment [AF]

Miscellaneous

Does any of the above help with a practical solution to ELK?

I think ELK is a promising approach to a subcomponent of alignment. This is because

1. It seems to circumvent some of the trickier concepts in the alignment space like agency or defining human values. This makes it much more probable to find a feasible solution within time.
2. It is applicable to many different possible architectures and approaches to powerful AI. It only assumes that the approach uses SGD which is not very restrictive (and isn't even a very crucial assumption for ELK IMO).
3. It feels like a huge step in the right direction. A solution to ELK would provide a mechanism by which we can understand the "true beliefs and intentions" of the underlying model and thus make much more informed predictions about the model's future actions and respective consequences.
4. In many ways, ELK seems like a minimal feasible solution to (a component of) alignment, i.e. it doesn't guarantee alignment but it makes most catastrophic failure modes much less probable if applied correctly.

Unfortunately, it's currently not clear how to make ELK work in practice. ARC is cautiously optimistic, that they can find a solution and reported some progress in their first report but translating that into a practical solution is still hard.

I think findings from Science of DL could yield valuable insights to speed up ELK. In particular

1. A possible implementation of ELK uses debate between two reporter heads on the weights/activations of the original network. The core bottleneck for this approach is scalable interpretability. Thus, providing better solutions to scalable interpretability would lead to progress in ELK.
2. The current assumptions of ELK are not very restrictive, e.g. the networks are trained with SGD. If we knew more about specific relevant architectures, e.g. how concepts are learned in transformers, we might be able to implement less general versions of ELK for these specific architectures. This wouldn't provide a general solution to ELK but making specific yet powerful architectures like transformers more secure would already be a huge step in the right direction.
3. One core contribution of ELK is to specify a training process that results in secure behavior. Understanding the inductive biases of DL or specific architectures better could allow us to narrow down the space of possible training processes for ELK and thereby make ELK easier.

Related work:

- Original ELK report [LW], [my summary]
- ELK prize competition [LW]
- ELK tag on LW [LW]

- Second ARC report [not yet public]

Investigating Causal understanding of LLMs

There are some people within the alignment community and in the broader ML community that think causality is either an important missing piece or "the holy grail" of ML. This ranges from the belief that Deep Learning can't learn causal models of the world to very specific and narrow claims about the necessity to understand counterfactuals better for alignment (Tom Everitt's group at DeepMind is working on this).
I think that DL can learn a causal understanding of the world in the same way that it can learn to add two numbers. If the model size, data and training process allow for it, an LLM will learn accurate causal models of the world. However, I think the argument that LLMs currently might not learn accurate causal models of the world is plausible. We have argued why causality might be important for alignment in this blog post and done a shallow investigation into the quality of the causal models of GPT-3 here but there is a lot more work to be done.

By now, I think causality is a relevant property, but it is by no means the only one that matters. I think investigations into this work are likely going to be a part of the broader "understanding emergent phenomena in LLMs" investigation rather than an investigation in its own right.

General Deep Learning theory

I have some general questions about DL theory but I think some of these questions are already answered by other academics and might therefore not be neglected enough. But it would probably be useful to follow the academic literature and draw conclusions for AI safety research or reproduce the results to check whether they are actually true. Some of these questions include

- What's the state-of-the-art explanation for double descent?
- What's up with this re-basin hypothesis of NNs?
- What's up with model soup?
- When does a DL algorithm start planning, e.g. does GPT-3 "plan" in some sense? Seems to be a crucial question for many AIS problems, especially deceptive alignment.
- How and when do mesa-optimizers arise in DL?

AI forecasting work

I am working with Epoch and done small reviews for OpenPhil on AI forecasting in the past. I still think this work is very interesting and important and I intend to continue to contribute as much as I can. However, some of my key uncertainties regarding timelines, take-offs and risks are empirical at this point in time. For many of these questions, e.g. how steep algorithmic progress is, my uncertainty can be reduced by running a lot of experiments on current state-of-the-art models. Thus, being able to design and run actual experiments to inform this type of work would be very helpful. Danny Hernandez at Anthropic has done a lot of work that is

similar to the work I mean. In case you want a better understanding of what this entails, I recommend skimming his work. I'm also interested in "trying to figure out where AI is heading" more generally similar to some of Jacob Steinhardt's, Ajeya Cotra's, OpenPhil's, OpenAI's (policy/strategy team) and Epoch's work.

I will keep most of these ideas private for now but I can invite interested readers to a different doc with more details.