1

Conditioning Predictive Models: Risks and Strategies

Kate Woolverton
TODO

Jan 19, 2023

Abstract

Our intention is to provide a definitive reference on what it would take to safely make use of predictive models in the absence of a solution to the Eliciting Latent Knowledge[1] problem.

Furthermore, we believe that large language models can be understood as such predictive models of the world, and that such a conceptualization raises significant opportunities for their safe yet powerful use via careful conditioning—e.g. extracting AI safety research via predicting counterfactual human research output.

Unfortunately, such approaches also raise a variety of potentially fatal safety problems, particularly surrounding situations where predictive models predict the output of other AI systems, potentially unbeknownst to us. There are numerous potential solutions to such problems, however, primarily via carefully conditioning models to predict the things we want—e.g. humans—rather than the things we don't—e.g. malign AIs.

Furthermore, due to the simplicity of the prediction objective, we believe that predictive models present the easiest inner alignment problem that we are aware of.

As a result, we think that conditioning approaches for predictive models represent the safest known way of eliciting human-level and slightly superhuman capabilities from large language models and other similar future models.

Contents

* Authors are listed in contribution order. Thanks to Paul Christiano, Kyle McDonell, Laria Reynolds, Collin Burns, Rohin Shah, Sam Marks, Nicholas Schiefer, Ethan Perez, William Saunders, Evan R. Murphy, Paul Colognese, Tamera Lanham, Arun Jose, Ramana Kumar, Thomas Woodside, Jared Kaplan, Beth Barnes, Robert Krzyzanowski, and Andrei Alexandru for useful conversations, comments, and feedback.

1 Large language models as predictors

Suppose you have a very advanced, powerful large language model (LLM) generated via self-supervised pre-training. It's clearly capable of solving complex tasks when prompted or fine-tuned in the right way—it can write code as well as a human, produce human-level summaries, write news articles, etc.—but we don't know what it is actually doing internally that produces those capabilities. It could be that your language model is:

- a loose collection of heuristics,[1]
- a generative model of token transitions,
- a simulator that picks from a repertoire of humans to simulate,
- a proxy-aligned agent optimizing proxies like sentence grammaticality,
- an agent minimizing its cross-entropy loss,
- a deceptive agent trying to gain power in the world,
- a general inductor,
- a predictive model of the world,
- etc.

Later , we'll discuss why you might expect to get one of these over the others, but for now, we're going to focus on the possibility that your language model is well-understood as a **predictive model of the world.**

In particular, our aim is to understand what it would look like to safely use predictive models to perform slightly superhuman tasks[2]—e.g. predicting counterfactual worlds to extract the outputs of long serial research processes.[3]

We think that this basic approach has hope for two reasons. First, the prediction orthogonality thesis[3] seems basically right: we think that predictors can be effectively steered towards different optimization targets—though we'll discuss some of

[1]Though some versions of the loose collection of heuristics hypothesis are still plausible, at a bare minimum such hypotheses must deal with the fact that we at least know LLMs contain mechanisms as complex as induction heads[2].

[2]We discuss the particular level of "superhuman" that we think this approach can reach later . Notably, when we say "narrowly superhuman", we mean on the individual task level. The ability to repeatedly simulating worlds with many narrowly superhuman intelligences for long periods of subjective time does not move a system beyond narrowly superhuman intelligence itself.

[3]Note that these tasks are not *necessarily* performed in counterfactual futures, and could be in e.g. counterfactual presents or predictions of very different worlds.

the many difficulties in doing so in Section 2 . Second, we think there is substantially more hope of being able to inner align[4] models to prediction objectives than other sorts of training goals[5] due to the simplicity of such objectives, as we'll discuss in Section 4

In the rest of this section, we'll elaborate on what we mean by a "predictive model of the world."[4]

1.1 Eliciting Latent Knowledge's prediction model

In "Eliciting Latent Knowledge[1]" (ELK), Christiano et al. start with the assumption that we can "train a model to predict what the future will look like according to cameras and other sensors." They then point out that such a predictor only tells you what your cameras will show: if your cameras can be tampered with, this doesn't necessarily tell you everything you might want to know about the state of the world.

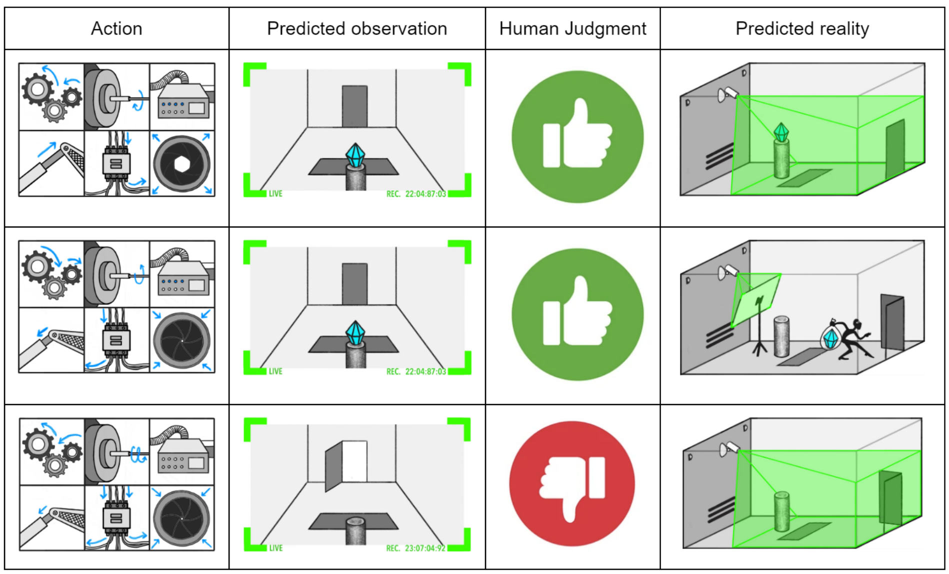

Figure 1: Above is the example given in the ELK report: if your predictor is only predicting what the camera shows, then you can't distinguish between a situation where the model predicts a thief will steal the diamond and put a screen in front of the camera and a situation where it predicts the diamond will just stay in the vault.

Such tampering becomes a serious problem if we directly use such a predictive model to do planning in the world—for example, if we always pick the action that is predicted to lead to the most happy humans, we could easily end up picking an action that leads to a world where the humans just look happy on the cameras rather than actually being happy. Christiano et al. propose solving this problem by attempting to access the predictor's *latent knowledge*—that is, its internal understanding of the actual state of the world.

Though we agree that using such a predictive model for direct planning would likely require accessing its latent knowledge to some degree, planning is only one of

[4]To keep our discussion general we'll assume that the model is multimodal, so it can also be conditioned on and output images/audio/video.

many possible uses for such a predictive model. Access to a model that we know is just trying to predict the future outputs of some set of cameras is still quite powerful, even if such a model is not safe to use for direct planning. This poses an important question:*is there anything that we could do with such a predictive model that would be both safe and competitive without being able to access its latent knowledge?* That question—or its equivalent in the large language model context—is the primary question that we will be trying to answer here.

To understand what sort of things we might be able to do with a predictive model, we first need to understand how such a predictive model might generalize. If we know nothing about our model other than that it was trained on a prediction task, there is nothing we can safely do with it, since it could have arbitrary behavior off-distribution. Thus, we'll need to build some conceptual model of what a predictive model might be doing that allows us to understand what its generalization behavior might look like.

Conceptually, we'll think of a predictive model as a sort of Bayes net[6] where there are a bunch of internal hidden states corresponding to aspects of the world from which the model deduces the most likely observations to predict. Furthermore, we'll imagine that, in the case of the ELK predictor, hidden states extend arbitrarily into the future so that the model is capable of generalizing to future camera outputs.

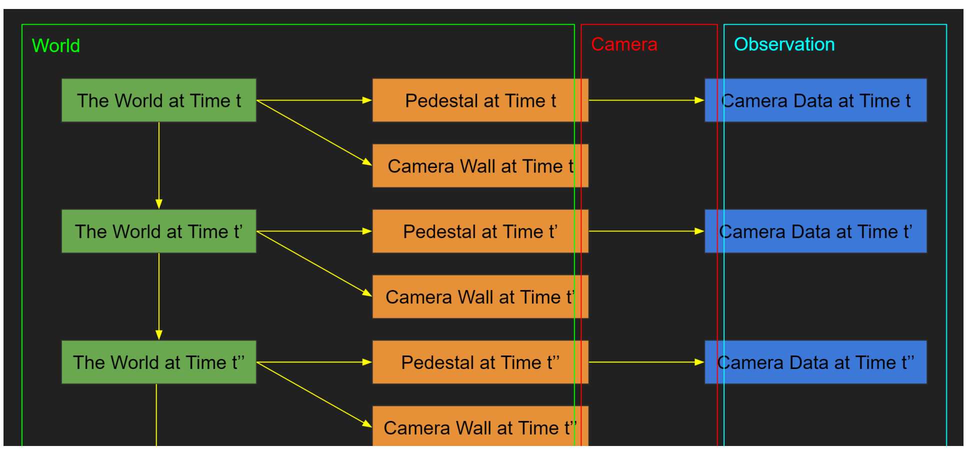

Figure 2: Our model of the ELK predictor. It has a bunch of internal states corresponding to aspects of the world, but its model of the camera only looks at some of those states such that only a subset influence the actual predicted observation. For example, the wall that the camera is mounted on is never observed.

Importantly, such a predictive model needs to model both the world and the camera via which its observations are generated from the world. That's because the observations the model attempts to predict are made through the camera—and because any other part of the world could end up influencing the camera in the future, so it's necessary for a good predictive model to have some model of the rest of the world outside of the camera too.

Additionally, such a model should also be able to accept as input camera observations that it can condition on, predicting the most likely camera observation to come next. Conceptually, we'll think of such conditioning as implementing a sort of back inference where the model infers a distribution over the most likely hidden states to have produced the given observations.

1.2 Pre-trained LLMs as predictive models

Though it might not look like it at first, language model pre-training is essentially the same as training a prediction model on a particular set of cameras. Rather than predict literal cameras, however, the "camera" that a language model is tasked with predicting is the data collection procedure used to produce its pre-training data. A camera is just an operation that maps from the world to some data distribution—for a language model, that operation is the one that goes from the world, with all its complexity, to the data that gets collected on the internet, to how that data is scraped and filtered, all the way down to how it ends up tokenized in the model's training corpus.

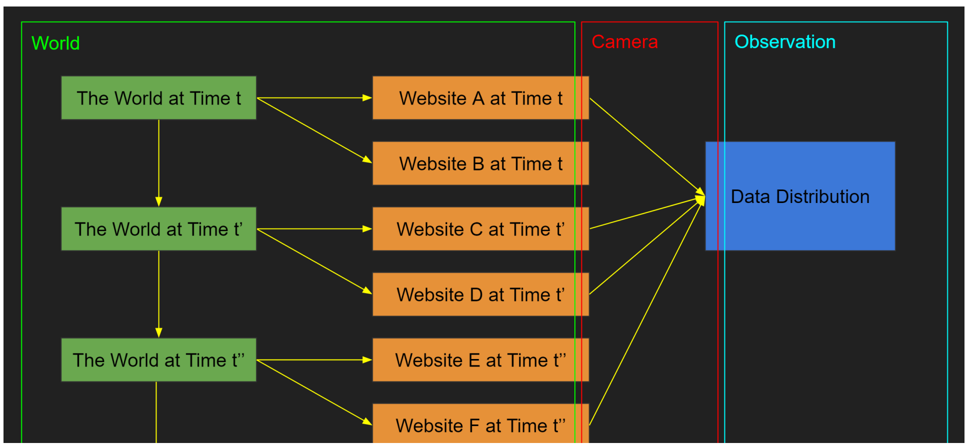

Figure 3: Our model of a pre-trained language model as a predictor. Such a model has to have hidden states corresponding to aspects of the world, be able to model how the world influences the internet, and then model how the internet is scraped to produce the final observation distribution that it predicts.

This analogy demonstrates that multimodal models—those that predict images and/or video in addition to text—are natural extensions of traditional language models. Such a model's "cameras" are simply wider and broader than those of a pure language model. Thus, when we say "language model," we mean to include multimodal models as well.

Importantly, the sorts of observations we can get out of such a model—and the sorts of observations we can condition it on—are limited by the "cameras" that the model is predicting. If something could not be observed so as to enter the model's training data, then there is no channel via which we can access that information.

For our purposes, we'll mostly imagine that such "cameras" are as extensive as possible. For example, we'll assume we can sample the output of a camera pointed at the desk of an alignment researcher, simulate a query to a website, etc. We don't think this glosses over any particular complications or roadblocks, it just makes our claims clearer.[5]

[5]We think that imagining arbitrary cameras is fine, since as long as we can build such cameras, we can just include their outputs as additional data in our training corpus.

There is one potentially notable difference between the LLM case and the ELK case, however, which is that we've changed our sense of time from that in the ELK predictor—rather than predicting future camera frames from past frames, an inherently chronological process, LLMs are trained to predict future tokens from past tokens, which do not have a strict sense of chronological order. We don't think that this is fundamentally different, however—the time at which the data was collected simply becomes a hidden variable that the model has to estimate. One difficulty with this handling of time, though, is that it becomes unclear whether such a model will be able to generalize to future times from training data that was only collected in past times. We'll discuss this specific difficulty in more detail in Section 2a

Language models have to be able to predict the world

We believe that language models can be well-understood as predictors in the sense that they have some model of how the world works from which they predict what their "camera" outputs would show.

Though there are many possible alternative hypotheses—which we will discuss in more detail in Section 4 —one particular common hypothesis that we think is implausible (at least as models get larger) is the hypothesis that language models simulate just a single actor at a time (e.g. the author of some text) rather than the whole world. This would suggest that language models only need to capture the specifics and complexities of singular human agents, and not the interactions and dependencies among multiple agents and objects in the environment.

The problem with this hypothesis is that it's not clear how this would work in practice. Human behavior isn't well-defined in the absence of an environment, and the text humans choose to write is strongly dependent on that environment. Thus, at least at a high level of capabilities, it seems essential for the model to understand the rest of the world rather than just the individual author of some text.

That said, we should not expect the model to necessarily simulate the entire world perfectly, as there are diminishing returns on token prediction accuracy with more world simulation. Instead, it seems likely that the model will simulate the immediate environment of the text-producing agents at higher fidelity, and more distant and less causally-connected aspects of the environment at lower fidelity.

The power of conditioning

Language models provide another mechanism of interaction on top of pure prediction: conditioning. When you prompt a language model, you are conditioning on a particular sequence of tokens existing in the world. This allows you to sample from the counterfactual world in which those tokens make it into the training set. In effect, conditioning turns language models into "multiverse generators[7]" where we get to condition on being in a branch where some set of tokens were observed and then look at what happens in those branches.

Furthermore, though it is the primary example, prompting is not the only mechanism for getting a conditional out of a large language model and not the only mechanism that we'll be imagining here. Fine-tuning—either supervised or via reinforcement learning (RL) with a KL penalty[8]—can also be used to extract conditionals, as we'll discuss later in Section 5 . Thus, when we say "conditioning," we do not just mean "prompting"—any mechanism for producing a conditional of the pre-trained distribution should be included.

In any situation where we are doing some form of conditioning, the multiverses we get to sample from here are not multiverses in the real world (e.g. Everett branches[9]), but rather multiverses in the space of the model's expectations and beliefs about the world. Thus, whatever observation we condition on, a good prediction model should always give us a distribution that reflects the particular states of the world that the model believes would be most likely to yield those observations.

An important consequence is that conditionals let us exploit hidden states in the dynamics of the world to produce particular outcomes. For instance, we can condition on an observation of a researcher starting a project, then output an observation of the outcome one year later. To produce this observation, the model has to predict the (hidden) state of the researcher over the intervening year.

Importantly, even though the dynamics of the world are causal, information we condition on at later times has effects on the possible world states at earlier times. For instance, if we know that we discover buried treasure in Toronto tomorrow, that heavily implies that the treasure was already there yesterday.

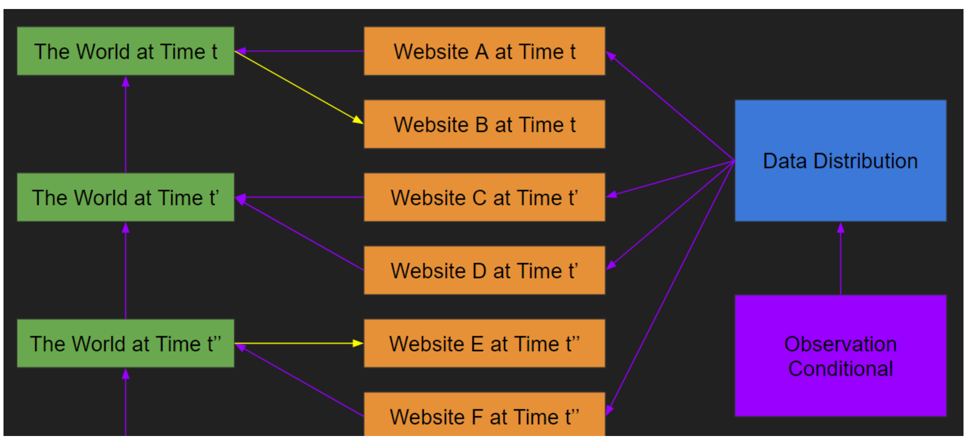

Figure 4: Our model of conditioning in language models. Observation conditionals lead to the model doing back inference to infer what states of the world would be most likely to produce that observation. Notably, the inference can only pass back through things that are directly observable by the model's "cameras."

While this is a powerful technique, it is nontrivial to reason about how the world will evolve and, in particular, what the model will infer about the world from the observations we condition on. For example, if the model doesn't know much about Evan Hubinger and we condition on it observing Evan move out of the Bay Area, it might infer it's because Evan wants to go home to his family—but that's just because it doesn't know Evan grew up in the Bay. If it knew quite a lot about Evan, it might instead infer that there was an earthquake in the Bay, since earthquakes are highly unpredictable sources of randomness that even a very advanced prediction model would be unlikely to anticipate.

Importantly, the conditionals that we get access to here are not the sort of conditionals that Eliciting Latent Knowledge[1] hopes to get. Rather than being able to condition on actual facts about the world (Is there a diamond in the vault?), we can only condition on observations (Does the camera show a diamond in the vault?)—what we'll call an *observation conditional.* That means that when we talk about conditioning our model, those conditionals can only ever be about things the model can directly observe through its "cameras," not actual facts about the world. Our ability to condition on actual world states entirely flows through the extent to which we can condition on observations that imply those world states.

It is worth pointing out that there are many kinds of conditionals that we expect to be useful but which are difficult to impose on current models. For example, we might want to condition on a news article being observed at nytimes.com, rather than just saying "New York Times."[6] Since we're trying to look forward to future models, we'll assume that we can access essentially arbitrary observational conditionals unless there is a clear reason to expect otherwise.

1.3 Using predictive models in practice

It is worth noting that the picture described above—of a model capable of conditioning on arbitrary observations and making accurate predictions about the world given them—is quite a sophisticated one. In our opinion, however, the sophistication here is just a question of the accuracy of the predictions: simply having some model of the world that can be updated on observations to produce predictions is a very straightforward thing to do. In fact, we think that current large language models are plausibly well-described as such predictive models.

Furthermore, most of our focus will be on ensuring that your model is *attempting to predict the right thing.* That's a very important thing almost regardless of your model's actual capability level. As a simple example, in the same way that you probably shouldn't trust a human who was doing their best to mimic what a malign superintelligence would do, you probably shouldn't trust a human-level AI attempting to do that either, even if that AI (like the human) isn't actually superintelligent.

That being said, the disconnect between theory and practice—the difference between a predictive model with perfect predictions and one with concrete capability limitations—is certainly one that any attempt to concretely make use of predictive models will encounter. Currently, we see two major approaches that machine learning practitioners use to attempt to bridge this gap and increase our ability to extract useful outputs from large language models:

1. fine-tuning with reinforcement learning[10] (specifically RL from human feedback) and

2. chain of thought prompting[11] (or other sequential reasoning techniques).

[6]As we'll discuss in Section 2a , we can get a better conditional here if the training data comes with metadata (e.g. URLs), but even then the metadata itself is still just an observation—one that could be faked or mislabelled, for example.

We think that both of these techniques can be well-understood under the predictive modeling framework, though—especially in the case of RLHF (reinforcement learning from human feedback)—we are uncertain whether predictive modeling is the best framework. Later in Section 4 we'll discuss in detail the question of whether RLHF fine-tuned models will be well-described as predictive.

In the case of sequential reasoning techniques such as chain of thought prompting, however, we think that the predictive modeling framework applies quite straightforwardly. Certainly—at the very least by giving models additional inference-time compute—sequential reasoning should enable models to solve tasks that they wouldn't be able to do in a single forward pass. Nevertheless, if we believe that large language models are well-described as predictive models, then trusting any sequential reasoning they perform requires believing that they're predicting one or more trustworthy reasoners. That means you have to understand what sort of reasoner the model was attempting to predict in each individual forward pass, which means you still have to do the same sort of careful conditioning that we'll discuss in Section 2 . We'll discuss more of the exact details of how sequential reasoning techniques interact with predictive models later as well.

1.4 The basic training story

"How do we become confident in the safety of a machine learning system?[5]" proposes the use of *training stories* as a way of describing an overall approach to building safe, advanced AI systems. A training story is composed of two components: the *training goal*—what, mechanistically, we want our model to be doing—and the *training rationale*—how and why we think that our training process will produce a model doing the thing we want it to be doing. We'll be thinking of the approach of conditioning predictive models as relying on the following training story.

First, our training goal is as follows: we want to build purely predictive models, as described above. That means we want to make sure that we aren't building models that are, for example, deceptive agents[12] pretending to be predictors. Furthermore, we'll also need it to be the case that our predictive models have a fixed, physical conceptualization of their "cameras."

In Section 2 , we'll discuss the challenges that one might encounter trying to safely make use of a model that satisfies these criteria—as well as the particular challenge that leads us to require the latter criterion regarding the model's conceptualization of its cameras. In short, we think that the thing to do here with the most potential to be safe and competitive is to predict humans doing complex tasks in the absence of AIs either in the present or the future. In general, we'll refer to the sorts of challenges that arise in this setting—where we're assuming that our model is the sort of predictor that we're looking for—as *outer alignment* challenges (though the technical term should be training goal alignment [5]), we think outer alignment is more clear as a term in this setting).[7]

[7]This usage of inner and outer alignment is somewhat contrary to how the terms were originally defined[4], since we won't be talking about mesa-optimizers here. Since the original definitions don't really apply in the predictive models context, however, we think our usage should be relatively unambiguous. To be fully technical, the way we'll be

Second, our training rationale: we believe that language model pre-training is relatively unlikely to produce deceptive agents and that the use of transparency and interpretability may be able to fill in the rest of the gap. We'll discuss why we think this might work in Section 4 . These sorts of challenges—those that arise in getting a model that is in fact a predictor in the way that we want—are the sorts of challenges that we'll refer to as *inner alignment* challenges (technically training rationale alignment[5]).

Furthermore, in Section 3 , we'll discuss why we think that this training story is competitive—that is, why we think such models will not be too much harder to build than plausible alternatives (training rationale competitiveness[5] or *implementation competitiveness*) and why we think the resulting model will be capable of doing the sorts of tasks we'll need it to do to in fact result in an overall reduction in AI existential risk (training goal competitiveness[5] or *performance competitiveness*). We'll continue this discussion as well in Section 6 when we look at what it might actually look like to use a powerful predictive model to reduce AI existential risk in practice.

2 Outer alignment via careful conditioning

Suppose we actually get a predictive model of the world that we can condition on arbitrary observations. What should we do with it?

One thing that is often done with large language models is to ask them to act as dialogue agents (e.g. predict what comes next after an "AI:" prompt). As we'll discuss extensively in this section, however, we think that asking a predictive model to predict itself or another AI system is highly unsafe. Instead, our basic proposal is to try to predict the most useful possible *human* outputs.

One canonical thing that you might want to do in that vein is "predict the results of humans doing AI safety research" (though note that this is just an example). Specifically, here's GPT-3 ('text-davinci-003') at zero temperature (prompt is bolded):[8]

> **# A full solution to the AI alignment problem**
>
> Unfortunately, there is no single solution to the AI alignment problem. AI alignment is a complex and multifaceted problem that requires a comprehensive approach.
>
> The first step in solving the AI alignment problem is to define the goals of the AI system. This involves understanding the desired outcomes of the AI system and how it should interact with its environment. Once the

using inner and outer alignment most closely matches up with the concepts of *training goal alignment*[5] (for outer alignment) and *training rationale alignment*[5] (for inner alignment).

[8] No cherry-picking was done for any of the samples we give here—each was our very first attempt to get the model to produce the behavior we were looking for. Though note that we did used to have an example here where we added a future date rather than adding "ARC Technical Report," which we then later removed following our additional investigations into conditioning on future dates that convinced us that the effect wasn't clearly real.

goals are defined, the next step is to develop a framework for how the AI system should behave in order to achieve those goals. This framework should include ethical considerations, such as ensuring that the AI system does not cause harm to humans or other sentient beings.

Unfortunately, as can be seen from GPT-3's continuation, it doesn't actually think a paper titled "A full solution to the alignment problem" will actually contain such a solution—instead, it seems to think it's far more likely that it'll just talk about what such a solution might entail. Which in fact seems true, at least if such a paper was written today. Fortunately, at least for current InstructGPT[13] models, it's not that hard to get GPT-3 to at least seemingly give a real attempt.[9] Here's another trajectory from 'text-davinci-003' at zero temperature (prompt is again bolded):

> **# ARC Technical Report: A full solution to the AI alignment problem**
>
> ## Introduction
>
> The AI alignment problem is one of the most pressing challenges of our time. It is the challenge of ensuring that artificial intelligence (AI) systems are designed and deployed in a way that is beneficial to humanity. This is a difficult problem because AI systems are often designed to optimize for a single goal, such as maximizing profits or minimizing costs, without considering the broader implications of their actions. As AI systems become more powerful and more widely deployed, it is increasingly important to ensure that they are designed and deployed in a way that is beneficial to humanity.
>
> This report presents a full solution to the AI alignment problem. It begins by discussing the current state of the problem and the challenges that must be addressed in order to achieve a full solution. It then outlines a comprehensive approach to solving the problem, including a set of principles and strategies for designing and deploying AI systems in a way that is beneficial to humanity. Finally, it discusses the implications of this approach and the potential for further research.

In this case—when we ask for specifically an ARC technical report—we at least seemingly get the model's actual best attempt at a solution rather than an attempt to sidestep it. Apparently, GPT-3 seems to believe that, if a document claims to be an ARC technical report, it'll be more likely to (or at least claim to) fully solve alignment. Of course, therein lies the key issue with all observation conditionals—we know we have now elicited a prediction of an article that claims to solve alignment, but we don't know whether that's actually the model's best attempt at such an alignment solution. Additionally, interpretation of what's happening here is made

[9] We did also try some of these prompts with base 'davinci' (only pre-trained, no fine-tuning), but it is quite difficult to get sensical outputs at temperature zero for 'davinci', as 'davinci' on temperature zero tends to produce a lot of repetition and memorized sequences. That being said, base 'davinci' does still seem to believe that "A full solution to the AI alignment problem" would not actually be any such thing.

more complicated by the fact that the model in question is a fine-tuned model rather than a pre-trained model—though as we discuss in Section 5 , we think it is plausible that many fine-tuned models, not just pre-trained models, can be well-understood as predictive models as well.

Regardless, the point is that, by conditioning on situations where the research produced by humans would be better, we can get the model to produce at least seemingly better outputs. And there are other sorts of potentially even more powerful observation conditionals we might be able to use here as well.

For example, we could try getting the model to predict what alignment research we would do in the future—presumably, future human alignment research will be substantially more advanced than current alignment research. At least for current models, however, getting actual predictions about the future is quite difficult, as they tend to instead simply predict what an article from the internet now but claiming to be from the future would say instead. We'll discuss this potential issue in more detail shortly , though note that it isn't necessarily a problem: predicting the future invites a great deal of peril, since the future could contain malign agents the outputs of which we'd rather not look at. Thus, it might be preferable to just predict counterfactual situations that might arise in the present instead.[10] Generally, we are relatively agnostic on the question of whether predicting counterfactual present worlds or future worlds will turn out to be superior.

Overall, as we'll discuss in this section, we think that there are a number of challenges in conditioning models to safely produce powerful alignment research, though we are optimistic that many of them at least have potentially promising directions for solutions.

It's worth pointing out, however, that doing alignment research is just an example task here: we think that the task of "do good alignment research" contains the necessary hard parts to serve as a useful example. While we also believe that using such a model for alignment research is a pretty reasonable thing to try, especially early on as a way to augment existing alignment researchers, we certainly don't believe that alignment research is the only possible use case for conditioning predictive models. Rather, we think that the same sorts of problems and techniques that we'll be discussing here for doing alignment research should apply equally to any other equivalently hard task that you might want to safely condition a predictive model into doing.

2.1 Should we predict the present or the future?

As we discussed previously, if we could get generalization to the future—e.g. by conditioning on something like "what will be on the Alignment Forum in 2050"—it could be a very powerful thing to do with predictive models. In such a situation, the model has to infer a plausible world trajectory leading to this observation and,

[10] Such a restriction to only predict counterfactual present worlds might both hamper competitiveness and invite a separate safety problem, however, due to the fact that very advanced research happening now is quite unlikely, so conditioning on it might make your model suspicious that something different than what you want—e.g. a malign AI, as we'll discuss in more detail later in this section —was responsible for generating the data.

for example, show us what it thinks alignment researchers are most likely to have written by then. Thus, we could use predictive models to let us pull forward insights from the future, which could be a promising way to accelerate alignment progress.

To be able to get actual predictions about the future when we ask these sorts of questions, however, we need our model to be able to predict a future trajectory if the conditional is more likely to occur in the future than the present. However, one plausible alternative is a model that takes the conditional and simulates a counterfactual *present* satisfying the conditional, regardless of how unlikely the conditional is to be satisfied in the present relative to the future.

The basic problem is that it is not clear that a model which has never seen future data would ever predict the future, at least without further interventions training it to do so. In the case of e.g. conditioning on a full solution to AI alignment, such a model might reason that it is more likely that somehow such a solution was produced now than that a solution from the future ended up in the model's training data. That being said, pre-training generally uses articles from a wide variety of different dates, so it should not be that implausible from the model's perspective that its training corpus might be extended with future data as well.[11]

However, our best guess is that current models don't generally make real predictions about the future, and instead mostly simulate counterfactual presents instead, though it is hard to get direct evidence about which of these possibilities is true without good interpretability tools. Anecdotally, GPT-3 seems to write from the standpoint of a human today rather than trying to predict an authentic future text, even when explicitly instructed otherwise—e.g. it tends to use "today" to refer to the present rather than the future date in question. We also have some preliminary indirect evidence that the model does not believe that text from the future is authentic. When asked to judge the authenticity of excerpts from newspaper articles, 'text-davinci-002' often (though not always) judges posts dated past 2022 as inauthentic, even when it judged them as authentic when dated prior to 2022, and even when the articles don't have obvious dating context clues (e.g., references to events in a particular year). This effect is not monotonic in time though: the same article might be judged as inauthentic in 2050, but authentic again in 2100. Moreover, the model is clearly not picking up on some obvious authenticity cues (e.g. an article on Democrats' success in midterms is judged as authentic even in odd-numbered years where there are no midterm elections, an article on snowy weather in New York is judged as authentic even in July, etc.). Though we weakly believe GPT-3 thinks text about the future is fiction, this definitely bears more study.

Fundamentally, the question of whether models will actually attempt to predict the future or not depends on exactly how they end up conceptualizing their "cameras"—for example, if some website that will exist in the future would, by future dataset collection procedures, be included in future training data, would the model include that in the distribution it's predicting? Presumably the model must at least

[11] Additionally, it should not be at all implausible that we would take a model pre-trained on data before some time and then fine-tune it on data from some future time, so for a model that interprets its "cameras" as tracking that sort of fine-tuning data, having future data points be from the future would be quite plausible.

have some uncertainty over when exactly the collection of its training data must have stopped.[12] If the model learns to model a camera with a strict time bound, however, that could make it difficult to access conditionals beyond that time bound.

Overall, we think that this sort of problem could make accessing predictions about the future quite challenging to do, at least by default. As we stated previously, however, that's not necessarily a bad thing: a lot of the problems that we'll talk about in the rest of this section are problems specifically because of how weird the future can be—it can contain lots of powerful AIs with lots of different potentially malign motivations, for example.

Nevertheless, there are a couple of reasons that we think predicting the present rather than the future could be a problem.

First, it presents a competitiveness problem: if you can only access conditionals about the present, you can only ever ask the model to do things that could plausibly happen now—otherwise it'll just predict that it won't happen. Even if your model has the capability to perform some task at a superhuman level, you won't be able to elicit that capability just via predictions about the present if there are no actors in the present that would ever plausibly be able to do that task at that level.

Second, even if you can somehow elicit the capability just via conditionals over the present, doing so could be even more dangerous than predicting the future, since it posits a very unlikely conditional that could cause the model to believe that something very strange was going on. For example: a solution to the alignment problem is very unlikely to happen now, which means for it to happen, the model might believe something very strange had to have occurred—e.g. maybe all the humans were actually tricked by some malign superintelligence, a failure mode we'll talk about in more detail in Section 2c .[13] One potential way to resolve this problem, however, could be to remove the information from the pre-training data that is causing the model to think that the conditional is implausible—e.g. filtering out anything contradictory—effectively implementing the desired conditional in the pre-training data itself. For that to work, however, the model has to not just start predicting your filtering process.

Regardless, we think it's quite likely that models can simply be trained in such a way that they will generalize to the future, even if they don't do so by default.

First, we could just train the model on time-ordered data such that at every point in training, the model has only seen past data and needs to predict future data—essentially recreating the ELK predictor setup. Thus, when we give it future short-term prediction tasks, those tasks are isomorphic to the tasks that it faced

[12] As a trivial example of a situation that a predictive model might think it's in that should lead it to predict future data relative to when its pre-training corpus was generated, it could be that a model pre-trained on 2021 data is then fine-tuned on actual 2022 data.

[13] It's possible that the model could conclude that other (unspecified) parts of the world are different in ways that make this less strange. For instance, it could infer that when we condition on extremely unusually good alignment research in the present we are asking about a counterfactual world where alignment research is further along, and produce predictions for that counterfactual world. That could mitigate these concerns, and put a floor on the improbability the model assigns to such scenarios (and hence a ceiling on how weird the predictions get).

during training.[14]

Second, another potential solution to this issue would be to train on text augmented with metadata, such that we can explicitly condition on the date in the metadata being from the future. For example, if all text in the training corpus is uniquely marked with the text's date and URL, we could explicitly condition an article as being observed to be from a reliable source in the future.[15] This way, when conditioning on e.g. an Alignment Forum post from the future, we are more likely to get the model's best attempt at such a post instead of a post from the present claiming to be from the future.

One thing that is a bit tricky with metadata conditioning, however, is that the model may still have some uncertainty over the metadata if it expects that there might be some mislabeling—but hopefully we should be able to use conditionals to provide the model with enough bits of evidence that it believes it's really predicting alignmentforum.org (or whatever else we want). Furthermore, it might be quite important for us to pre-commit to always make any metadata included in training as genuine as we can—and to do so in a way that is as obvious to the model as possible—so as to convince it not to ever predict that future metadata was intentionally mislabeled. Notably, it should still be fine to test/prompt/deploy with mislabeled metadata, as long as mislabeled metadata never appears in training, since ideally the model should only learn a "camera" pointed at the training data collection procedure, not any test data generation procedure.[16]

Overall, we are generally agnostic with respect to whether predicting counterfactual present worlds or future worlds is better from either competitiveness or safety perspectives.

References

[1] Paul Christiano, Mark Xu, and Ajeya Cotra. Eliciting latent knowledge: How to tell if your eyes deceive you. 2021. URL `https://www.alignmentforum.org/posts/qHCDysDnvhteW7kRd/arc-s-first-technical-report-eliciting-latent-knowledge`.

[2] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly,

[14] A concern with this approach is that either the model only sees a very small number of examples of future prediction (if the time-ordered data have significant gaps) or else all of its examples will have very small time intervals between what it has seen so far and the next prediction. This is not necessarily a problem for longer-term prediction, as a model that learns to robustly predict a minute into the future is something we can likely iterate to predict much further, but it does make it seem less likely that the model straightforwardly generalizes to the task of predicting over longer time-horizons.

[15] Probably the ideal way of including metadata would be to insert `<begin metadata> url=... date=... <end metadata>` tags (probably with url and date in a random order) at random points in the text, thus allowing you to always be able to extract the model's beliefs over the metadata by conditioning on `<begin metadata>` (where `<begin metadata>` is some special token). Credit to Kyle McDonell and Laria Reynolds for suggesting this particular implementation.

[16] It is possible for the model to end up predicting the test data if the model simply learns to predict any data stream anywhere in the world that looks like the data it has seen so far. In that case, even mislabeled metadata at deployment time might be a problem. We discuss further what sort of "camera" the model might learn in Section 4 .

Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads, 2022. URL https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

[3] Janus. Simulators. 2022. URL https://www.alignmentforum.org/posts/vJFdjigzmcXMhNTsx/simulators#:~:text=In%20general%2C%20the,the%20model%E2%80%99s%20power.

[4] Evan Hubinger, Chris van Merwijk, Vladimir Mikulika, Joar Skalse, and Scott Garrabrant. Risks from Learned Optimization in Advanced Machine Learning Systems. *arXiv*, 2019. URL https://arxiv.org/abs/1906.01820.

[5] Evan Hubinger. How do we become confident in the safety of a machine learning system? 2021. URL https://www.alignmentforum.org/posts/FDJnZt8Ks2djouQTZ/how-do-we-become-confident-in-the-safety-of-a-machine.

[6] Bayesian network, January 2023. URL https://en.wikipedia.org/w/index.php?title=Bayesian_network&oldid=1134656115. Page Version ID: 1134656115.

[7] Janus. Language models are multiverse generators. 2021. URL https://generative.ink/posts/language-models-are-multiverse-generators/.

[8] Tomek Korbak and Ethan Perez. Rl with kl penalties is better seen as bayesian inference, 2022. URL https://www.lesswrong.com/posts/eoHbneGvqDu25Hasc/rl-with-kl-penalties-is-better-seen-as-bayesian-inference.

[9] Evan Hubinger. Multiple worlds, one universal wave function, 2020. URL https://www.lesswrong.com/posts/2D9s6kpegDQtrueBE/multiple-worlds-one-universal-wave-function.

[10] Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. Technical Report arXiv:1706.03741, arXiv, July 2017. URL http://arxiv.org/abs/1706.03741. arXiv:1706.03741 [cs, stat] type: article.

[11] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. Technical Report arXiv:2201.11903, arXiv, January 2023. URL http://arxiv.org/abs/2201.11903. arXiv:2201.11903 [cs] type: article.

[12] Evan Hubinger. How likely is deceptive alignment?, 2022. URL https://www.alignmentforum.org/posts/A9NxPTwbw6r6Awuwt/how-likely-is-deceptive-alignment.

[13] Aligning Language Models to Follow Instructions, January 2022. URL https://openai.com/blog/instruction-following/.